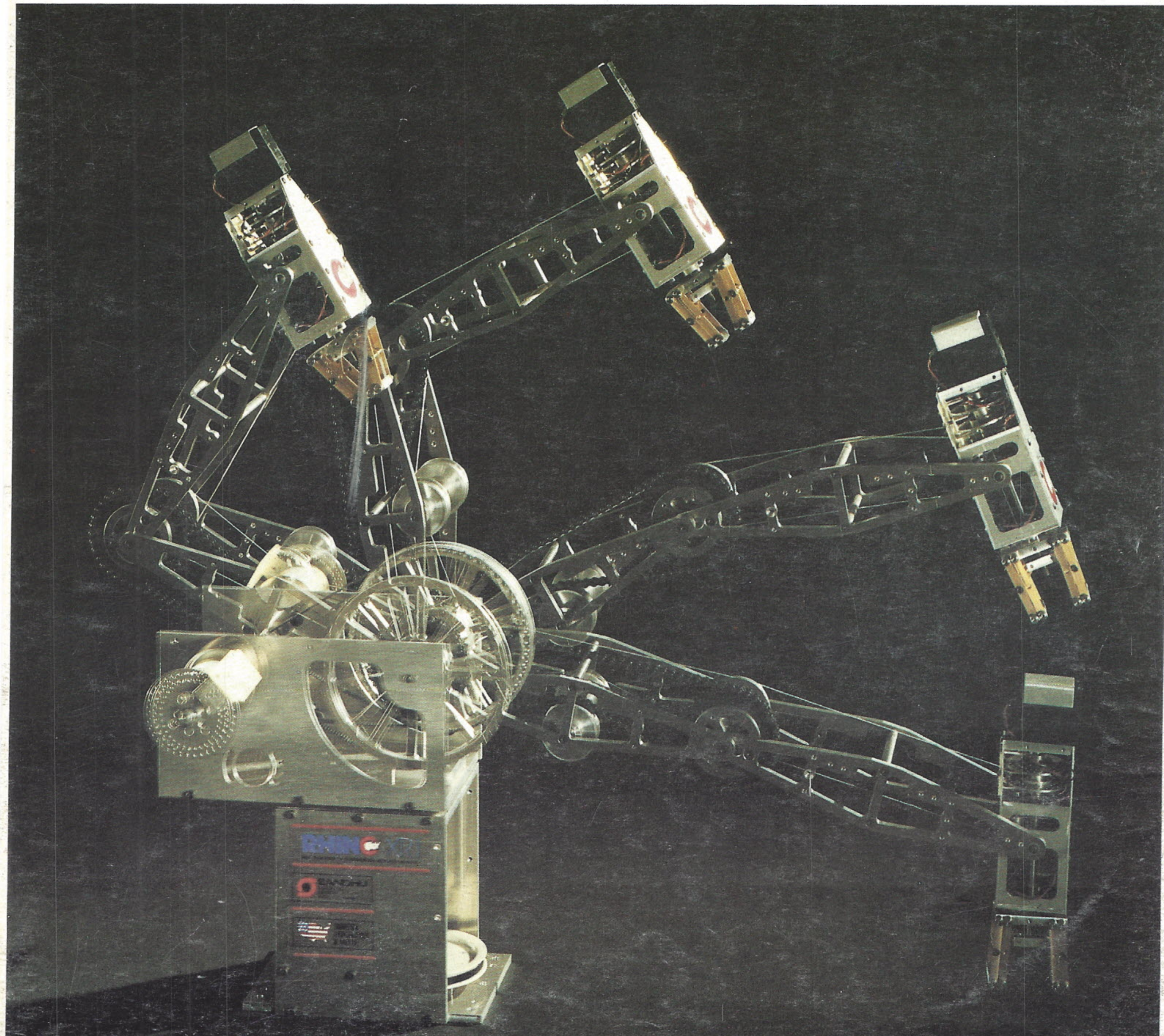
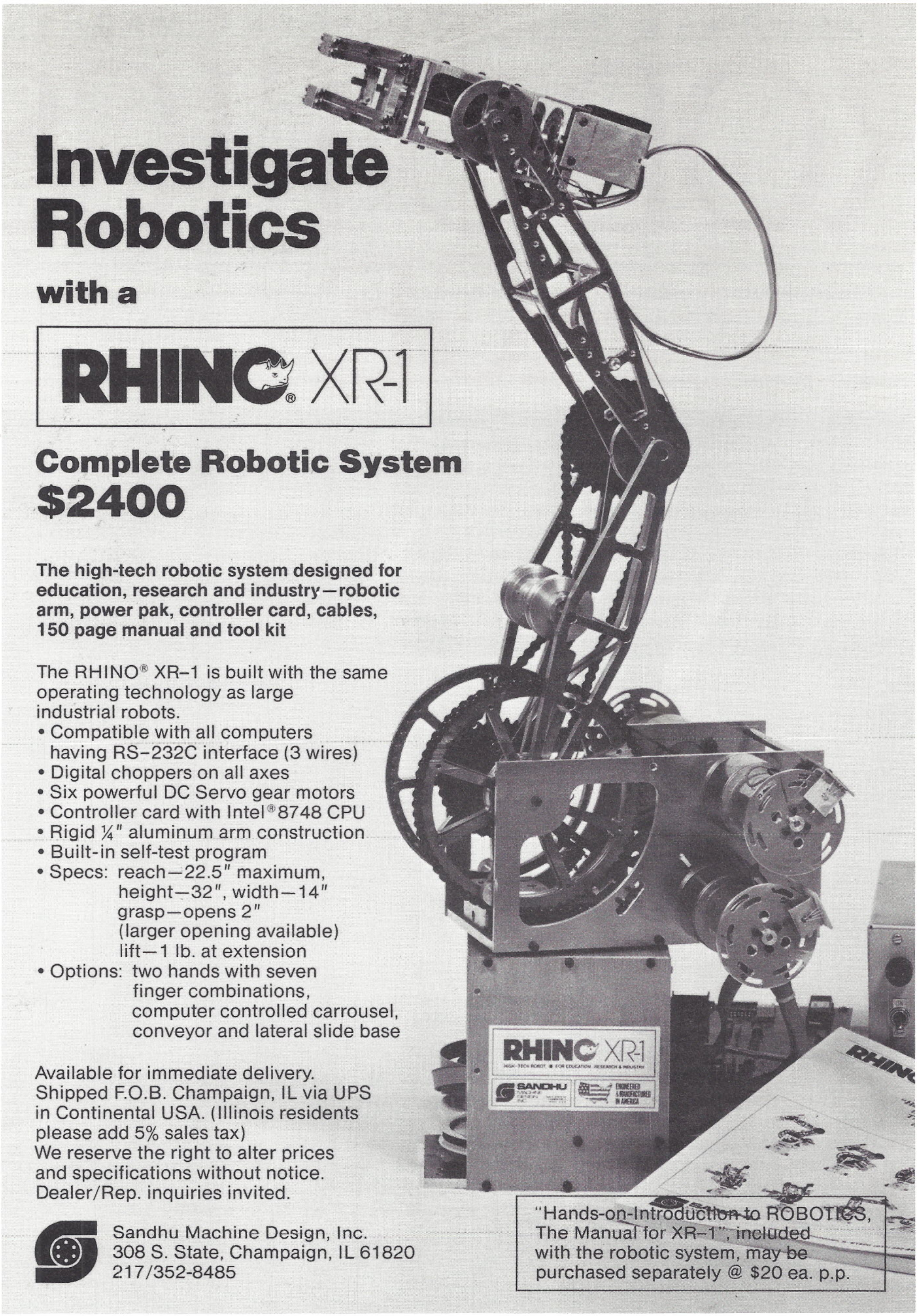
MAR/APR 1982 VOL. 4 NO. 2

$3.00 USA/$3.50 CANADA

THE JOURNAL OF INTELLIGENT MACHINES

ROBOTICS AGE™




Investigate Robotics
with a
RHINO® XR-1
Complete Robotic System
$2400
The high-tech robotic system designed for education, research and industry—robotic arm, power pak, controller card, cables, 150 page manual and tool kit
The RHINO® XR-1 is built with the same operating technology as large industrial robots.
• Compatible with all computers having RS-232C interface (3 wires)
• Digital choppers on all axes
• Six powerful DC Servo gear motors
• Controller card with Intel® 8748 CPU
• Rigid ¼" aluminum arm construction
• Built-in self-test program
• Specs: reach—22.5" maximum, height—32", width—14" grasp—opens 2" (larger opening available) lift—1 lb. at extension
• Options: two hands with seven finger combinations, computer controlled carrousel, conveyor and lateral slide base
Available for immediate delivery. Shipped F.O.B. Champaign, IL via UPS in Continental USA. (Illinois residents please add 5% sales tax)
We reserve the right to alter prices and specifications without notice.
Dealer/Rep. inquiries invited.
Sandhu Machine Design, Inc.
308 S. State, Champaign, IL 61820
217/352-8485
RHINO XR-1
SANDHU
"Hands-on-Introduction to ROBOTICS, The Manual for XR-1", included with the robotic system, may be purchased separately @ $20 ea. p.p.

CIRCLE 12

CIRCLE 7

CIRCLE 17

THE JOURNAL OF INTELLIGENT MACHINES

ROBOTICS AGE™

MAR/APR 1982 VOL. 4 NO. 2

EDITORIAL

FEATURES

About the Cover: The cover of this issue is a photograph of the Rhino XR-1 arm which was provided by Sandhu Machine Design, Inc. In the photo we see a multiple exposure effect which emphasizes the motion of the mechanism.

DEPARTMENTS

ROBOTICS AGE — USPS 523850 (ISSN 0197-1905) is published six times a year in January, March, May, July, September and November by Robotics Age Inc., Strand Building, 174 Concord Street, Peterborough, N.H. 03458, phone (603) 924-7136. Address subscriptions, change of address, USPS Form 3579, and fulfillment questions to *Robotics Age* SUBSCRIPTIONS, P.O. Box 358, Peterborough, N.H. 03458. Second class postage paid at Peterborough N.H. and at additional mailing offices.

Subscriptions are $15 for one year (6 issues), $28 for two years (12 issues), $39 for three years (18 issues) in the USA and its possessions. In Canada and Mexico, subscriptions are $17 for one year, $32 for two years, $45 for three years. For other countries, subscriptions are $19 for one year surface delivery. Air delivery to selected areas at additional charges, rates upon request. Single copy price is $3 in the U.S., $3.50 in Canada and Mexico, $4 in Europe, and $4.50 elsewhere. Foreign subscriptions and single copy sales should be remitted in United States funds drawn on a U.S. bank.

Address all editorial correspondence to the Editor, at *Robotics Age*, Strand Building, 174 Concord Street, Peterborough, N.H. 03458. Opinions expressed by the authors of articles are not necessarily those of *Robotics Age*. To aid in preparation of acceptable articles, the *Robotics Age* Authors' Guide is available upon request if accompanied by a self-addressed 8½ inch by 11 inch envelope with first class postage for 3 ounces. Unacceptable manuscripts will be returned if accompanied by a self-addressed envelope with sufficient first class postage. Not responsible for lost manuscripts or photos.

The Intelligent Machines Industries: A View of the State of the Art

Carl Helmers

The intelligent machines industries are new. This newness is not tied to an exact date of birth, but rather to the growing realization of their importance. Recent developments in the history of technology have helped spur their birth as a body of thought and practices.

In electronics we think of the progression from the first transistor to today's large scale integration of 64K bit chips, or the 16-bit microcomputer's processor that is contained on a single chip. In mechanical fields we think of the continuing improvements in areas ranging from materials engineering and manufacturing arts to the interfaces of electronic and mechanical systems. In the aerospace field, the use and design of intelligent machines are universal. At one end of the spectrum we think of the modern digital autopilot in its commercial and experimental forms; at the other end we think of the autonomous robot cruise missiles and their peaceful counterparts in interplanetary space probes. In the world of consumer goods we see wonders ranging from electronic games to self-diagnosing automobiles, intelligent kitchen and household appliances to personal computers and calculators.

There is a common thread which binds all these technological trends together — the use of computers to implement the artificial approximations of intelligent behavior demanded by these real world systems. Computers and their applications are at the heart of these new industries. The challenge is to utilize inexpensive modern computer powers in cost effective and innovative ways. Computers have rapidly broken out of the conventional mold of blithe and innocuous data processor. They have entered the real world of designs once confined to science fiction dreams.

An Informal Survey

The intelligent machines industries are really a collection of tools, practices, and design approaches involving application of computer systems techniques. The industries we include in this area of computer application are several. All involve automation. All involve use of computers and software techniques to achieve specific applications of general purpose computing elements.

Manufacturing

Machine intelligence is becoming essential in the manufacturing industries. There the word "robot" has become a recent code word for manufacturing automation that uses flexible, reprogrammable manipulators. But intelligent machines in manufacturing are hardly limited to manipulators and the automated production line: the whole area of computer-aided design and sophisticated computer-aided manufacturing requires the application of intelligent machine engineering disciplines.

The products that result from the manufacturing uses of machine intelligence vary. The purpose, improving and enhancing manufacturing productivity, remains the same in all cases. The computer-aided design installation provides software and tools that enhance the ability to create manufacturable designs. The numerically controlled machine tool can take the instructions from that design facility and build the tooling necessary to produce parts from the design. The robot arm manipulator can then be employed in numerous tasks that use the tooling in a production process. At all levels of the process, machine intelligence in the form of software for computer systems is a key element.

The economic justifications of manufacturing automation developments are obvious and unassailable: the newer techniques can result in real productivity and cost improvements with very short payback periods. The inherent charisma of such automation is the tantalizing prospect of the totally automated factory, the ultimate capital good. The results impact on all other areas of human endeavor.

Editorial

Consumer Products

It's one thing to apply the intelligent machine concept to the process of manufacturing. Going one step further, we begin to see more and more use of machine intelligence in the objects being manufactured.

In the late 1970's we saw microprocessor controlled sewing machines, microprocessor controllers for microwave ovens, and all manner of intelligent toys. In the 1980's the first big orders were made for microprocessors as intelligent system controllers in automobiles. And toys have certainly become much more sophisticated.

The ultimate consumer application of intelligent machines is the domestic servant robot. The more "practical" skeptics would rightly respond "show me." We can't predict when, but with a past record of turning science fiction into technology fact, it must happen. A whole cottage industry of tinkerers and experimenters is already at work trying to perfect prototypes of this ultimate appliance.

Civilian and Military Aerospace Products

Intelligent machine concepts have been part of the aerospace electronics design field for the past several decades, sometimes evolving with the technology, and sometimes being forced to evolve by aerospace applications of the technology. The latest in flight instrumentation for military and civilian aircraft epitomizes the use of contemporary intelligent machine design.

An airline pilot once described to me the sensation of flying in the cockpit of a Boeing 747, a design that is more than a decade old. In effect, he said that "you dial in the numbers, sit back, and relax while the plane flies itself from New York to London." While that statement is somewhat exaggerated, the trend is quite real. There are inertial navigators and satellite navigation systems of unprecedented accuracy. Cockpit automation computer systems planned for the next generation of planes allow use of two-man crews. And there are existing instrument landing systems and projected collision avoidance systems that will greatly improve the safety aspects of flying.

Then there's the defense industry's latest — the cruise missile — a much improved version of the World War II German "buzz bomb" system. This autonomous flying robot has but one purpose — reaching its target under active self-generated guidance to deliver a bomb. But the same class of algorithms that makes this weapon so effective in the face of hazardous terrain has peaceful uses as well. We will eventually be able to teach an automobile to drive from point A to point B, saving lives that would otherwise be lost.

Natural Resource Recovery

The intelligent machine is a major factor in the recovery of increasingly scarce resources from our planet's mineral trove. We can save lives and lower costs by using a sort of intelligent teleoperator for mining activities. This, however, is only an immediate and obvious use.

The abstract field of artificial intelligence seems far removed from the physical reality of such projects, yet it has already entered into the field of resource recovery. Significant work is under way in the area of "expert systems," specifically as underwritten by oil exploration budgets. The analysis of a complex spectrum of data taken from seismic prospecting is ripe for automation, and "expert" analysis of oil field data could greatly improve the process of exploration.

Analysis of earth resources data with software that employs scene recognition, image enhancement, and other techniques of computer-aided analysis is still another area. Expert systems research in artificial intelligence is not confined to one field of expertise. It borders on the general theory of knowledge representation and the act of getting to information in databases.

Research

All of this leads up to an important "application area" of the intelligent machine engineering culture — research into the limits. As with all frontier activities, the border between generally accepted engineering practices and the wild land of new techniques is continually being pushed back. Research into the use of computer sensing, planning, and control is one of the most important areas of future applications of the technology.

Research into the limits of intelligent machine design can take many forms. At one level it is the amply funded research of the professional working in the context of a manufacturer, an industrial research organization, academic institution, or a government agency. At the other end, it is the scantly funded innovation of the imaginative tinkerer with a personal computer, a knowledge of electronics, and a willingness to experiment with a particular application.

Everyone has had the opportunity to read science fiction. Few have had a chance to implement it. The exploration of new functions in a research environment is a major source of the excitement of the field. Getting a computer program to play at grand master chess levels is a feat of machine intelligence. Having a mobile robot map its environment and feed itself is an act of machine intelligence that marks a great accomplishment for the machine's designer.

Exploring new uses of robotics in factory, laboratory, even domestic projects is a reward unto itself. The designers and innovators of the industries that are just now in the early stages of existence recognize the opportunities, as intelligent machines become an ever more important part of every day existence.

Media Sensors

conducted by Scott Nowell

Articles of Interest. The number of articles on robots is becoming too large to allow detailed summaries, so I will list some of our article sources and give some brief summaries and reviews.

Most major newspapers and new magazines are printing articles on robots, robot technology, and the effects of robots on our lives and in our business. These articles add credence to the inevitability of robots in our future, and it's good to know that there is a growing public awareness of robots. The theme of most articles in the general press, however, seems to revolve around how robots are affecting Japan, and their rapid acceptance of them. Hopefully, this media coverage will kindle the interest of the general public to the point where we too have a factory with robots building robots.

An excellent source for finding articles on robots in the general magazine press is the *Readers Guide to Periodical Literature*. This wonderful guide can be found in almost every library, and it lists references to almost any subject. From the *Readers Guide to Periodical Literature*, Volume 81 Number 10, August 1981, come the following references:

"Robots: A Growing Maturing Population," W.J. Cromie, *Sci-Quest*, March 1981.

"Tools and Monsters," M.C. Miller, *New Republic*, May 16, 1981.

"Unicorn-1 Robot," J.A. Gupton, Jr., *Radio-Electronics*, January, March, April, May 1981.

"New Smarts on the Production Line," L.A. Phillips, *Technology Review*, May/June 1981.

"Robots at the Ready" (Cincinnati Milacron), *Fortune*, June 1, 1981.

The *Readers Guide to Periodical Literature*, Volume 81 Number 17, December 1981, yields:

"Look, No Hands" (Yamazaki Machinery Works Computer Controlled Plant In Japan), *Time*, November 16, 1981.

Most of the articles we report on are ones we have either read, or found out about from readers. We can't possibly see them all and welcome any assistance. If you wish to contribute an article so we can tell the rest of the readership, send the article, a photocopy, or just drop us a card telling where to find it. If you send a copy, be sure to include the publication name and the issue date, so we can give credit to the original publisher. We are looking for any article dealing with robots, robot technology, how robots are being used, or how robots do or might affect us. In addition, we are looking for information on new products or any general news about the industry. Send your contributions to: "Media Sensors," *Robotics Age* magazine, Strand Building, Concord St., 202N, Peterborough, NH 03458.

Recent Articles. *Material Handling Engineering* magazine is an excellent source of information on the robot industry. Their industry will be one of the first to be affected by robot technology, so they do a comprehensive job of reporting.

As expected, *Material Handling Engineering* contains many articles and news items about robots. In the January 1982 "Computer/Controls/Components" column, editor Gene Schwind discusses some of the problems of marking and recognizing containers and container handling. In "the eyes of the robot," he says, "the question is not whether automatic identification is here to stay, but what form it will finally take." The main choices now are optical recognition or symbol scanning.

Joe Quinlan describes the XR-6100 robot from PaR Systems, a division of GCA Corporation, Bedford, Massachusetts, in "The Leading Edge" section. The robot is an electro-mechanical unit mounted on a telescoping mast that rides on a rail-mounted trolley.

"Soviets Reveal Targets For Space Program," *Electronic Engineering Times*, November 9, 1981. A piece entitled "The Future of Soviet Space Exploration," released by the Novosti Press Agency in Moscow, has defined the goals for the Soviet space program. A quote from the Novosti release reads, "Soviet specialists believe that the most promising direction in the development of automatic systems is the designing of efficient self-contained robots, capable of traveling across the surface of planets, of perceiving and analyzing the environment, and of deciding on a course of action once the situation is analyzed. The designing of such automatons will depend on a solution to the problems of artificial intelligence and integral work."

"The Robots Are Coming And Japan Leads Way," Hidehiro Tanakadate,

Media Sensors

U.S. News and World Report, January 18, 1982. Two pages are devoted to the increasing uses of robots in factories and numerous examples of actual situations in Japan are given. The main emphasis is on the business side of the robot revolution and why so many companies are becoming involved. In a quote from the Matsushita Company, Akira Nagano, a spokesman, says, "In our color-TV plant, five component-insertion robots, tended by four workers, can do the same amount of work formerly done by 44 humans." The robots cost from $31,000 to $132,000, work two shifts a day, and the workers who were replaced are drawing salaries and fringe benefits totaling about $790,000 a year.

"Japan's Love Affair With the Robot," Henry Scott Stokes, *The New York Times Magazine*, January 10, 1982. Complemented by nine color photographs, this article describes different uses of robots in Japan and explains why the Japanese are so enthralled. It also explains why Japan is doing so well with the robot revolution, but has a human interest angle. In an interesting quote, Roger B. Smith, chairman of General Motors, says, "Every time the cost of labor goes up $1 an hour, 1,000 more robots become economical."

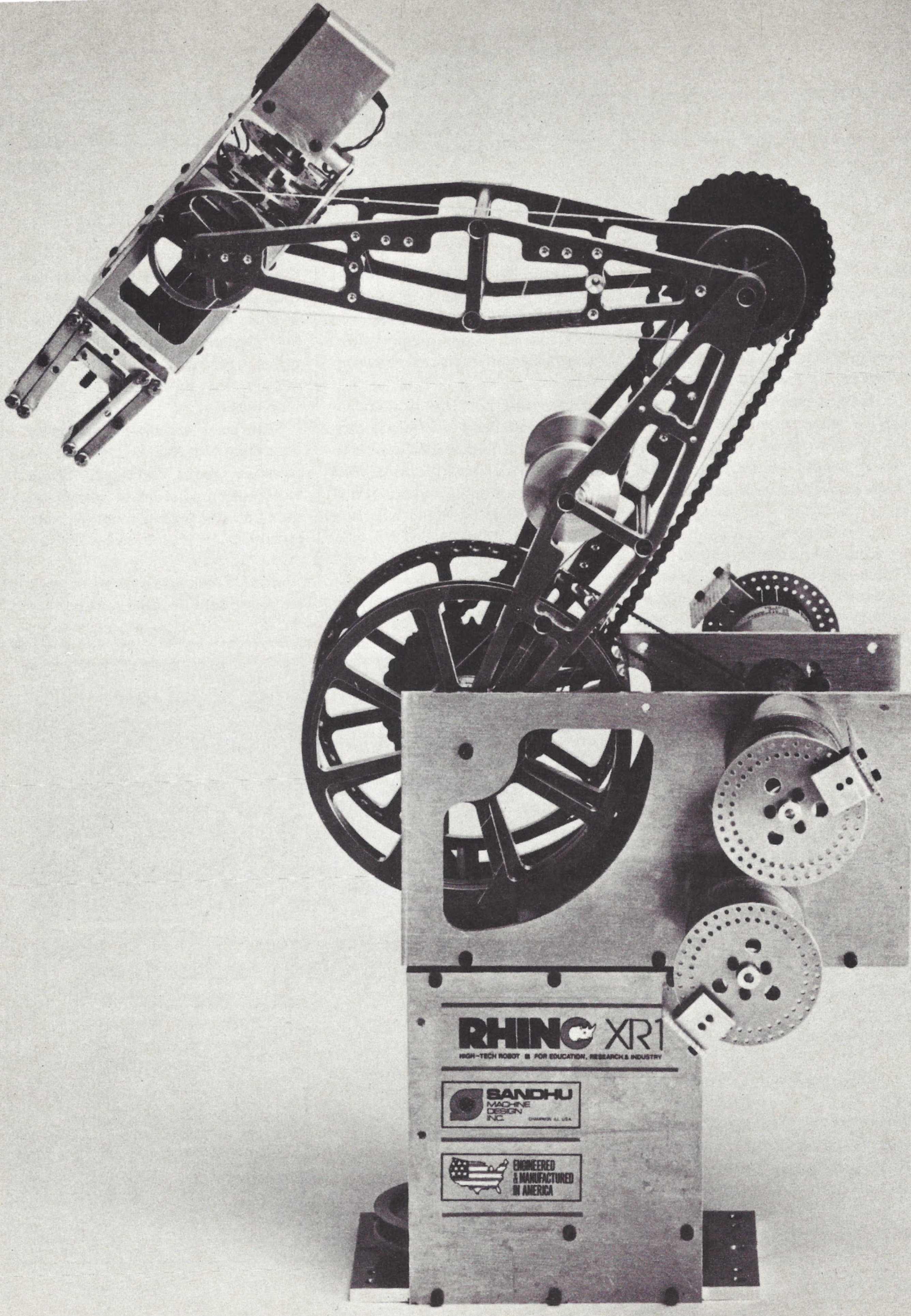
"Automaking Robots Pose Problems," Howard Bierman, Electronics Review-Robotics, *Electronics Magazine*, January 27, 1982. At least one of the big-three automakers in the United States is having its share of troubles with the introduction of robots into its assembly lines. Dan Kuchens, an electrical design engineer at Ford Motor Company's Milpitas, California, plant said that robots are still not ready for prolonged periods of work in a factory environment.

Based on a panel discussion on industrial robots at a technology conference sponsored by the New York-based Financial Analysts Federation that took place early in January, the article continues by saying that some of the difficulties posed by $100,000 per robot installations are horrendous. While many problems can be fixed in as little as a half hour, this can lead to labor-relations problems for top management because the robot must be temporarily replaced with a human welder. Training assembly-line workers to work with robots poses problems, because they view robots as a threat to their employment.

Bierman sums up his article by describing the intense Japanese interest in robots, and says, "Some forecasters predict Japan will have as many as one million robots in their assembly plants by the year 2000."

"Some Corporate Giants Are Rushing Into Robots," *Business*

Continued on Page 53


RHINO XR1
HIGH-TECH ROBOT ■ FOR EDUCATION, RESEARCH & INDUSTRY
SANDHU
MACHINE
DESIGN
INC.
ENGINEERED
& MANUFACTURED
IN AMERICA

THE RHINO XR-1

A Hands-On Introduction to Robotics

Harprit Sandhu

Sandhu Machine Design, Inc.

308 S. State

Champaign, Illinois 61820

Opportunities for investigations into the field of robotics have until now been limited either to large research facilities and industrial and educational laboratories, or the small hobby shops in the basements of avid tinkerers. With the introduction of the Rhino XR-1, anyone interested in robotics can have access to a five-axis manipulator that can be completely controlled from any small computer. Hardware, firmware and software are modifiable.

The Rhino XR-1 is specifically designed as a research and education tool for the industrial environment. Modeled after the large industrial robot arms, it is completely digital in design and features information transfers from the computer to the Rhino XR-1 and from the Rhino XR-1 back to the computer. A controller card with an on-board microprocessor handles all the overhead associated with the control of the motors; the control, drive, and logic circuitry needed to monitor the entire robot are all on this one card. The card can also monitor six separate interrupts that can be configured as desired.

Origin

The Rhino XR-1 is a product of Sandhu Machine Design Inc., a firm in Champaign, Illinois, that specializes in the design and fabrication of special machinery. As one might expect, engineers everywhere who are associated with design machinery talk about the "robot revolution," but at our shop, the discussions led to the decision that we would design and market a small, sophisticated robot arm.

The first concrete opportunity to put our plan into effect arose early in 1981. The resources were committed and the result was the Experimental Robot-One, known as the XR-1. The Rhino concept comes from its strength and ruggedness.

Photo 1: The Rhino XR-1 stands 32 inches high and contains six motors, one for each of the five axes of the arm-like mechanism and a sixth to control the fingers. Using four basic commands, the Rhino can be programmed to perform a variety of sophisticated tasks.

Early Specification Ideas

Before a machine can be designed, its personality has to be defined. The following describes the initial personality of the Rhino XR-1. It is not necessarily what we ended up with, it is simply how we began.

It was decided that the Rhino XR-1 should be controlled by any computer with an RS-232C interface, the most common computer interface at this time. This means that, to the computer, the robot is a serial ASCII device no different from a terminal, modem, or printer.

It was further decided for reasons of economy that the full RS-232C interface should not be required. The robot would be designed so that it could be controlled by three wires. The three signals used are:

- Transmitted data, Line 2 (transmitted by the XR-1).
- Received data, Line 3 (received by the XR-1).
- Ground, Line 7. Common data ground.

It should also be possible, we decided, to run Rhino XR-1 from a modem in case someone needed remote capability.

We also wanted a data transmission rate that was adjustable from 300 to 9600 bps. Of these:

- 300 bps gives adequate operation. This rate is used by slow modems.
- 1200 bps is much better. This rate is used by the more expensive modems.
- 9600 bps gives crisp operation. Used for direct connection to a computer on site. This rate is available on almost all computers and is thus more flexible than 19200 bps.

We thought it advisable for the robot to have its own controller card. Since we'd decided on the RS-232C interface, this could make it possible to decrease the overhead on the host computer. Computers, being hundreds of times faster than the robots they control, could then be used for making further calculations and decisions while the robot's movements were being executed.

With this in mind, the Intel 8748 processor was chosen to control the interface, and design considerations were initiated.

We decided to avoid the use of stepper motors because of the following problems:

- Slippage and thus loss of positional information.
- Easy overloading of motors.
- Need for timed control at all times.
- Need for constant power for running and holding the motors.
- Ramping considerations are a must so that the inertial properties of the motors will not be overlooked.
- No longer digital if slippage occurs.

It was further decided that a good small robot should work just like the big industry robots. This would require that optical choppers be placed on all axes of the robot so that exact axes positioning could be repeated indefinitely. Eliminating loss of position information over a long period of time is very important for sophisticated operation and control.

As a rule of thumb, the resolution of the Rhino XR-1 would be about $\frac{1}{32}$ inch and the speed about 20 feet per minute or three seconds for every foot of movement (of the major axes under consideration). (Incidentally, the big computerized numerical control machines run at about 200 to 400 inches a minute when traversing a table — not unlike Rhino XR-1.)

Since the RS-232C interface can use 12V, it was decided that the motors should run at 12V. For the small amount of 5V power needed for the logic components, we used an onboard voltage regulator and stole the 5V from the +12V side of the power supply.

A bi-polar power supply was chosen because it eases the problems associated with reversing DC motors. The system is implemented on the controller in the following manner. One side of the motor is always grounded, the other side is then connected to either +12V or −12V as needed to run the motor forward or backward. The microprocessor makes sure that both cannot be on at the same time. Needless to say, there are other schemes for achieving this result, but this was the most cost effective.

We also decided, for the sake of simplicity, to attempt to construct the robot almost entirely out of $\frac{3}{8}$ inch, $\frac{1}{4}$ inch, $\frac{1}{8}$ inch, and $\frac{1}{16}$ inch aircraft-grade aluminum. We would keep the rod sizes to a few standard shafts around $\frac{1}{4}$ inch, $\frac{5}{16}$ inch, and $\frac{3}{8}$ inch, and keep fastener sizes to a minimum. No rivets would be used and everything would be designed for easy disassembly.

All of this amounted to a big complicated order. However, we decided to build an initial run of 100 robots. If the acceptance in the marketplace was good and there was a need for such a robot, we could easily produce 100 robots a month.

We also determined that the educational and experimental nature of the robot would necessitate that it be designed so that it could be completely disassembled by the purchaser and put back together again. A tool kit would have to be provided to facilitate this process.

There could be no easily breakable parts — they would have to be substantial and solid and designed for a long, useful life. The experimenter would be able to see everything: nothing would be hidden unnecessarily.

In order to keep the cost of the robot down, the use of elaborate bearings would be minimized: not that bearings themselves are expensive, but they are time consuming to fit and retain accurately and, thus, expensive. Our life tests on the robots indicated that if a drop or two of 3-in-1 oil is applied to lubricate each lube point occasionally, the machine should last indefinitely.

It would be unrealistic to expect the base of the robot to be sufficiently heavy to allow the robot to lift heavy objects with its arms fully extended, so it was decided that the base would be designed to be easily attachable to a work surface. (The robot should not be used unless the base is attached to a solid surface.) A hole, a slot, and a threaded hole would be provided for mounting. The robot arm would be built as a space frame which would be extremely light and extremely rigid.

We determined that the body and the base should be fairly simple structures and allowances made so that the belts that drive the various axes could be easily adjustable.

In addition, the unit would have good firmware transparency, so that the person programming the arm would not have to worry about the machine firmware when developing his or her own software.

Finally, the machine would have to be easy to use, easy to program, flexible and versatile, and above all, fun and easy to learn from.

Actual Design Parameters

The Rhino XR-1 robot arm consists of three main components and their related devices:

- the mechanical arm,
- the controller card,
- the power supply (including all related cables and devices).

The Mechanical Arm. The mechanical arm is 32 inches high, and has five degrees of freedom plus controllable fingers. The unit is configured as follows:

- A base upon which the main body of the robot arm rotates. The base contains the motor that rotates the robot arm about the waist and forms a platform for the body to be mounted upon. It may be mounted in any position, and should be bolted down.
- A body element that forms the main structure for the drive mechanisms and mounts the three motors that control the movement of the bicep, the forearm and the wrist azimuthal movement. This section of the device contains the drive mechanisms for the controls, gears, and belts needed to effect the motions of the arm assembly.
- The bicep and the forearm consist of space frames made of $\frac{1}{4}$ inch square aluminum carved out of $\frac{1}{4}$ inch plate and assembled with $\frac{1}{4}$ inch diameter spacers. These form

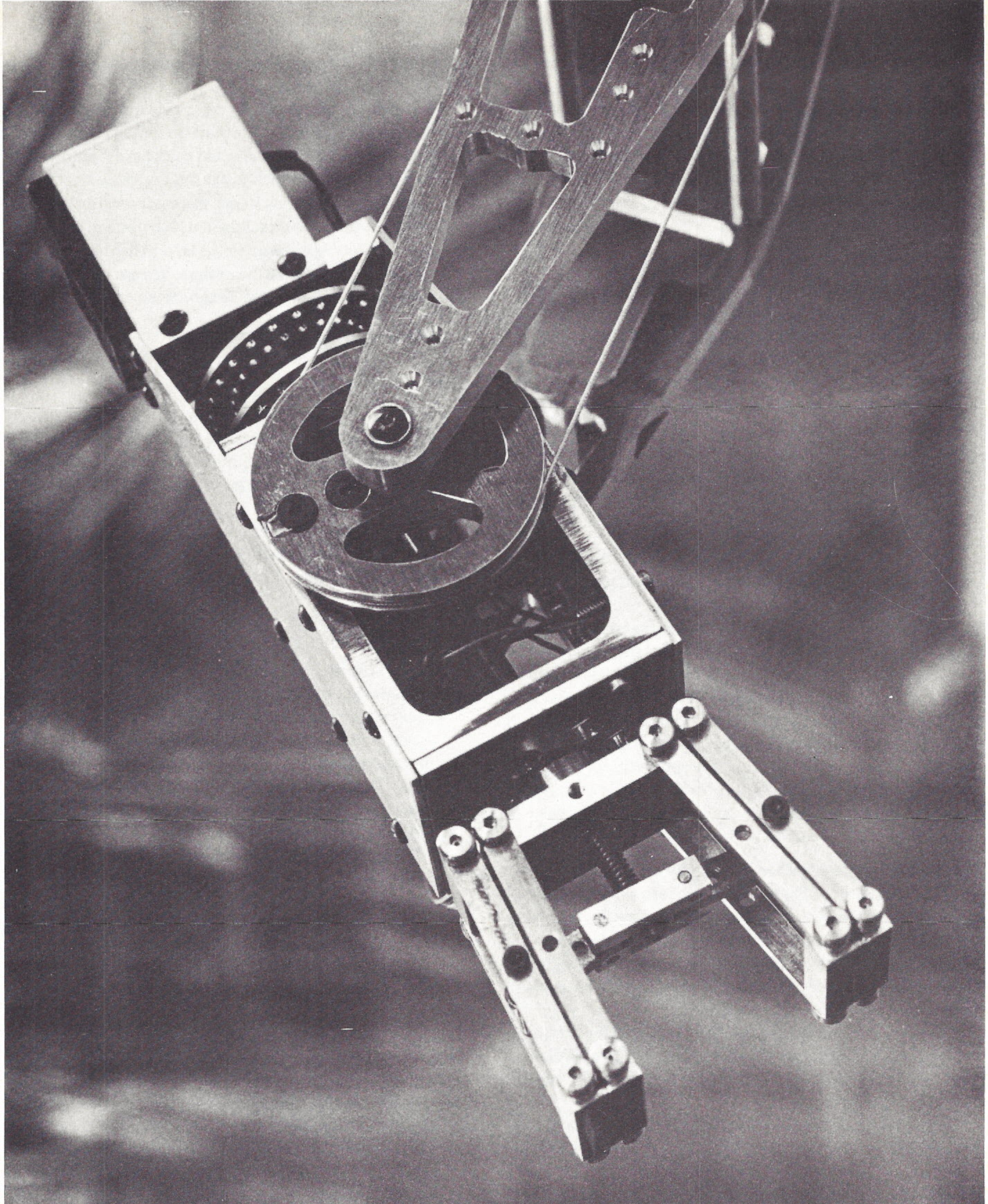


Photo 2: The Rhino's hand is mounted at the end of the forearm. The rotational movement of the hand and the closing of the fingers is controlled by two hand-mounted servomotors. The Rhino can reach up to 32 inches from base to top of fingertips and has a radial reach of 22.25 inches from center of rotation to tip of fingers. The arm's lifting ability and gripping force are approximately 16 ounces.

very rigid space frames and related hardware is mounted on them.

- The hand is mounted at the end of the forearm. The azimuthal movement of the hand is controlled from the body of the Rhino XR-1. The rotational movement of the hand and the closing of the fingers is controlled by two hand-mounted servomotors. One motor controls the rotation of the wrist and the other controls the closing of the fingers.

All six motors on the Rhino XR-1 have optical incremental encoders attached to them. This system of optical choppers provides a very effective form of digital positioning for each of the axes. The choppers for the waist, the shoulder joint, elbow joint, and wrist azimuthal joint run at the full speed of the motors before the motion has been transmitted through the gearbox. These motors can be positioned within $1/6$ of a revolution. The choppers for the two hand motors are mounted on the output shafts of the gearboxes and have a resolution of $1/24$ of a revolution or 15 degrees. (The deluxe hand provides a resolution of about $1/10$ of one degree!)

The Controller Card. The Rhino XR-1's controller card provides complete digital control of eight motors and monitors their optical choppers. It also has an on-board RS-232C interface that receives information from and sends information to the host computer. The ability to monitor the condition of six interrupts is implemented on the controller card: four of these interrupts are configured as microswitch monitors on the Rhino XR-1; the other two are available as needed. If necessary, all six interrupts can be made independent of the operation of the motors. Because the controller card provides such easy control of eight motors on a completely digital level, it can be used effectively by experimenters who wish to develop their own mechanical arms.

Operation of Controller Firmware

The program stored in the 8748 processor chip consists of a scanning routine. This scanning routine scans the host computer output present at the RS-232C interface and scans the operation of the optical choppers on up to eight motors. It also sends information from the Rhino XR-1 to the host computer when information is requested.

Six motors are on the robot. The basic package can control two additional motors. These motors may be coordinated with robot operation. This can be most useful and someday most robots will be able to do this.

The program maintains the motors positions by constantly monitoring the status of the two photo-transistors on each chopper. Essentially, the microprocessor determines the direction in which the motor is rotating and keeps track of how many holes have gone by on the chopper as the rotation proceeds. If the motor overshoots the required number of holes, the motor is automatically reversed and backed up the required number of holes. This provides digital damping. The controller attempts to constantly maintain a zero error position for each motor. This is done automatically and is transparent to the user. New commands are added to the error signal when received.

Should an extremely heavy load cause the arm to drop, the change in count will be picked up by the controller and the controller will move the affected motor in the correct direction to bring the count back to the zero error position. If you move a motor chopper by hand, the motor will automatically start and oppose the direction in which you are moving the chopper. Most observers are surprised at the force required to oppose the operation of the motors. We are using very high reduction ratios, and it is necessary to be careful.

Conversing With the Controller

The host computer can give the Rhino XR-1 controller four separate commands:

- START a specific motor and move it a specific amount. The amount is expressed as a number of chopper holes.
- QUESTION. What is a motor's error condition now? How far is it from its stopping point?
- STATUS. What is the status of the six microswitches (another name for interrupt lines)?
- STOP a specific motor instantly and set its error condition equal to zero.

All commands must end with a carriage return and line feed. The commands are used in the following way:

The Start Command. The start command consists of a two to five character string that is organized as follows:

a. The first character can be from A to H (upper- or lowercase) and identifies one of the eight motors that the card is controlling.
b. The second character can be a plus (+) or minus (−) sign and indicates the direction in which the motor is to run. The plus (+) sign may be omitted.
c. The last three characters can be anywhere from 0 to 127 and indicate the number of counts that the motor is to be moved. The 0 count conveniently defines a null instruction. The count of 95 is the maximum count that should be sent out at any one time. (It is actually possible to send 127, but it is not recommended unless you know exactly what you are doing. More on this later.)

Motor Status Information. The QUESTION to request motor status information from the controller is a two-character string. The first character is a letter from A to H and the second character is a question mark (?). With this instruction we are asking the computer if the motor under consideration has reached its zero error position. If that motor has reached a zero error position, the controller will answer with a hexadecimal 20. If the motor has not reached its zero error position, the character returned will be hexadecimal 20 plus the number of holes still to be traversed, all expressed in hexadecimal notation. Hexadecimal 20 is added to the error condition because some computers use all codes below 20 as control codes and this can cause problems. The information received is valid for a very short time since the motor is run-

ning at full speed. Making use of this information has to be done quickly and with some cunning. Essentially, this allows us to send the motor under consideration the next instruction without waiting for the motor to stop. It also allows us to inhibit the next instruction until the error count is low enough so that we will not go over the 127 count which would flip the direction bit.

The fact that hexadecimal 20, decimal 32, has been added to the count returned means that the highest count we can receive is 127 − 32, or 95. After that, the numbers wrap around and 96 is expressed as −32, 97 as −31 and so on to −1 which represents 127. If your computer will not accept numbers below hexadecimal 20 because it uses these numbers as control codes, then you have to limit yourself to keeping all conversations with the robot to numbers below decimal 95. If you never ask a motor to move more than 95 holes, it can never be more than 95 holes from its destination and you will never encounter control codes in your conversations with the Rhino controller card. Note that the wraparound gives you negative numbers from −1 to −32. On some computers, however, the (−) bit is not processed properly and this can cause problems.

Interrupt Status. The controller card has the ability to detect the grounding of six lines. These six lines are organized as the limit switch microswitches on motors C, D, E, F, G, and H, the six motors that allow full power to be applied to them (not the two small hand motors). As presently configured, these six lines are used to detect the microswitch closures for these six motors. They do not necessarily have to detect microswitch closures — they can be used to detect the grounding of these six lines for any purposes. The instruction for asking the status of these limit switches is "I". A question mark is not required. Upon receiving the "I" command, the Rhino controller card responds with a hexadecimal number. In decimal equivalents, if all the microswitches are open, you will get a 95 back from the controller. If any of the microswitches are closed, the number will change as follows:

- If switch C is closed, decrement by 1.
- If switch D is closed, decrement by 2.
- If switch E is closed, decrement by 4.
- If switch F is closed, decrement by 8.
- If switch G is closed, decrement by 16.
- If switch H is closed, decrement by 32.

Thus, if all the switches were closed, we would decrement by 1 + 2 + 4 + 8 + 16 + 32 = 63 and the answer would be 32. The number can never be below decimal 32 (hexadecimal 20), the magic number we wanted to stay above earlier.

Stop a Motor and Set Error Signal to 0. This is a two-character command. The first character consists of a letter from A to H and the second character consists of an X. Thus, if motor C is running or stalled under load and CX is sent to the controller, the controller will immediately set the error signal for motor C to 0 and that motor will be stopped. This

command is of particular interest as it allows us to turn off stalled motors and to effect other recovery procedures that are needed from time to time. This command can stop the hand closing motor and remember where the hand stalled on the object being handled.

These four commands constitute a comprehensive language capability between the host computer and the controller for the Rhino XR-1. A very sophisticated capability can be designed with these commands because they give complete control of the motors. The experimenter can design still other commands for his or her own firmware: your imagination is the only limit. The flexibility of the design that we have created is the key to allowing this implementation.

The Intel 8748

The controller design is based on the Intel 8748 microcomputer. The Intel 8748 is a new NMOS chip that allows the design and development of sophisticated control systems with relative ease. The chip consists of:

- 8-bit processor which allows powerful control.
- 1K by 8-bit erasable read-only memory. Ease of programming on-board.
- 64 by 8-bit programmable data memory which provides adequate programmable memory.
- 27 I/O lines, for lots of I/O capability directly from the chip.
- 8-bit timer/event counter. Not used in controller at this time but can be used for fantastic feedback possibilities from the Rhino XR-1. Can be used to ramp motors!
- Single 5 V power supply requirement for convenience.
- More than 90 instructions constitute a powerful instruction set.
- All packaged in 40-pin DIP package.
- Available in both an EPROM and ROM version for economy.

Firmware Operation

The on-board firmware is designed to perform two functions: to provide an RS-232C interface for conversations back and forth with the host computer, and to monitor the operation of eight DC motors and their optical choppers.

The microprocessor on board the controller monitors the situation at the eight DC motors as follows:

The two photo-transistors on the choppers at each of the motors are designed to provide two signals that are 90 degrees out of phase. The two signals are read by the microprocessor as digital signals that look like 00, 01, 11, 10, 00. . . when the motor is running in one direction, and look like 00, 10, 11, 01, 00. . . when the motor is running in the other direction. The microprocessor uses the generated sequences to determine the direction in which the motor is turning and to update the current status of the error signal for each of the motor positions.

When the motor reaches its zero error position, the microprocessor automatically turns the power off to that motor. Should the motor be disturbed by a load, or overshoot its destination, the microprocessor will immediately detect the change in the error signal and automatically take corrective action. Once the microprocessor has been told to move a motor a certain number of chopper holes, the execution of the command is completely automatic. The one exception is that an overload or stall condition will cause the microprocessor to keep trying to get to the zero error position. The QUESTION command will tell you when this stall condition is anticipated, and corrective action can be taken automatically under software control. The power supply is not overloaded. The motors can be protected.

For more information on the operation of the firmware, see *A Hands-On Introduction To Robotics, The Manual for the XR-1*.

A Complete Robotics System

The availability of interchangeable hands, additional fingers, and other accessories make the Rhino XR-1 a complete robotics system.

Hands. Two basic types of hands are available for the Rhino XR-1: the standard hand (comes with the Rhino XR-1), and the deluxe hand (currently available by special order, but in the future will be available off the shelf).

The standard or H-1 hand is the lighter and smaller of the two. It has two small plastic servomotors to power the rotation of the wrist and the closing of the fingers.

The deluxe hand is the same width as the standard hand, so it fits in the same space, but it has slightly more depth and less height. It has two all-metal DC gearbox motors that are smaller versions of the four larger motors that run the arm and base. These motors are arranged to give more torque, more power, and better resolution to the accessories that fit on the hands. The choppers run at full motor speed and provide extremely fine control.

Both hands use the same accessories and are interchangeable, except, of course, for the parts that constitute the hands themselves.

Fingers. A comprehensive family of fingers can be ordered, and they will be available off the shelf soon. These fingers and palms, which do the actual grasping of the various manipulated objects, are available in the following configurations:

The *basic fingers* (F-1) come standard with the Rhino XR-1. These consist of a two-finger arrangement in which the tips of the fingers stay parallel to one another. Other items can be attached to these flat fingertips to allow the hand to grasp a variety of objects.

The *1.5X fingers* (F-1.5) are 1.5 times larger than the standard fingers to allow slightly more flexibility and grasp of larger objects. The fingers come with their own wrist plate and nut and easily replace the standard fingers.

The *2X fingers* (F-2) have been increased in size by 2.0 times to allow even more flexibility in grasping larger objects. The fingers come with their own wrist plate and nut and easily replace the standard fingers. As might be expected, increas-

ing the size of the fingers decreases the force at the fingertips.

The *triple fingers* (F-3) are essentially the standard F-1 fingers but with three sets of fingers, which makes it a much better hand for grasping round objects. Additionally, objects tend to center themselves in the hand when grasped, and this is a tremendous plus when it comes to writing pick and place routines. The fingers come with their own tendon shaft, wrist plate, and nut, and easily replace the standard fingers. As with all finger accessories, these can be used on both hand mechanisms.

The *magnetic fingers* (F-MP) consist of a magnetic pickup unit that can be attached to the hand mechanism to pick up objects made of ferro-magnetic materials. There is no need for a tendon on this hand since no closing is involved. A dummy tendon shaft is supplied. Power to the hand is not supplied as such. One of the extra motor lines is selected to power the relay supplied to switch the hand. (Under limited conditions, it is possible for sophisticated users to use the controller power supply for weak applications.) At the owner's option, the magnetic hand can, of course, be hooked up to a separate, powerful power supply (recommended procedure).

The *clamshell fingers* (F-CS) consist of a small clamshell-type arrangement whose operation is similar to the basic hand supplied with the unit. Instead of fingers, however, this arrangement has a clamshell that allows the Rhino XR-1 to pick up objects that would otherwise be difficult to grasp.

The *shovel fingers* (F-SV) consist of a shovel-type hand. The tendon nut is not used as a closing device, but rather as a tilting device not unlike a digging bucket. A dummy tendon shaft is supplied with the unit.

The *hollow body hands* (F-MD) unit consists of a hollow hand body that takes the place of the standard hand and allows the user to position a small MotoDremmel (R) tool in the hand of the Rhino XR-1. This can be used for grinding and carving applications.

Sandhu Machine Design Inc. designs and builds other hands and robots to customer specifications. Please write for quotes. A sketch can be very helpful as can a sample of the object you want the hand to manipulate.

Other Accessories

The Rhino XR-1 system can be supplied with the following accessories, currently built to order, but available off the shelf in the near future.

Carousel. A small carousel that can be easily positioned at the front of the robot is provided as a convenience item to robotics investigators. The carousel consists of a platter to which various items can be attached. It rotates under command from the controller and can be connected to the seventh or eighth motor control point on the controller board. This allows the experimenter to perform complicated pick and place experiments on a moving carousel under the complete control of the host computer (without having to go through all the work that is normally required to hook up a separate carousel and attach switches and power supplies to it).

Conveyor. A small conveyor that can be positioned in front of the robot base is provided as an experimental accessory for serious experimenters. It can be used to simulate larger conveyor systems that might be encountered in an industrial environment. The conveyor is about three inches wide and about 18 inches long, so it is within easy reach of the robot at either end of its travel, and it is powered by a motor that can be run from port 7 or port 8 of the controller board under complete control of the host computer.

Linear Base. For the ambitious experimenter, we provide a linear base that allows the robot to be mounted so it can traverse laterally about 18 inches. The motor on this base can be run from port 6, port 7, or port 8 of the robot. If port 6 is used, the waist is disabled and lateral motion takes the place of the waist operation. If port 7 or port 8 is used, waist operation can continue along with lateral movement of the entire robot.

If the lateral moving option is used, special precautions must be taken to allow the controller and the power supply to follow the robot. These problems are left to the experimenter to solve, as Sandhu Machine Design Inc. does not currently provide any appropriate accessories (although if the demand is sufficiently high, we may provide these accessories in the future). The most economical solution is to use long

cables on the motors. These can be ordered from Sandhu Machine Design in lengths up to 10 feet.

Power Supply. A bipolar power supply is provided with the Rhino XR-1, and this allows the design of a simple reversing mechanism for the DC motors. This is effected by grounding one side of a motor and connecting either + 12 V DC or − 12 V DC to the other side of the motor.

The standard power supply provides 3.5 amps at each voltage. The minimal 5 V needs of the logic components are taken from the + 12 V side of the supply and regulated by an on-board regulator. This is adequate for most purposes, but if extremely heavy use is envisioned, a heavier power supply (7.0 amps) is available.

Experimentation

The Rhino XR-1 is designed to allow experimentation at three separate levels of interest: hardware, firmware, and software.

Hardware. The unit can be easily modified by the experimenter as construction is almost entirely of free machining aluminum that requires minimal shop facilities. The entire unit can be taken apart and assembled at will, and a tool kit is provided. The availability of the extra hands, fingers, conveyors, carousels, and lateral slide base allow still further experimentation.

The removal of one screw detaches the wrist and fingers from the hand. Another hand can be attached immediately and the experiment can proceed.

A motor can normally be detached with two screws. Should the experimenter choose to investigate different motor types, it would be a fairly straightforward matter to attach the new motor to the Rhino XR-1.

The optical encoder disks are attached with three screws, and should the experimenter wish to modify the encoding procedures, the process is simple enough.

And on and on!

Firmware. Detailed listings of the firmware that controls the controller card are provided. The serious experimenter can write new firmware to suit his or her own special interests. A completely new language may be developed, or new commands added. The firmware provided with the Rhino XR-1 gives the guidance you need to see how you might go about developing your own software. Adding to the software is fairly straightforward.

Software. Because the Rhino XR-1 can be run from an RS-232C port from any computer, software can be developed in any higher-level language or machine language. Conversations with the Rhino XR-1 are initiated with simple PRINT commands, none of which are more than five characters long. The host computer can think that the Rhino XR-1 is a serial ASCII printer or similar device. All responses from the Rhino controller are only 1 byte long, so no complicated busy signal and handshake logic is needed or implemented in the controller design. The processor is fast enough to keep up with all commands from the host computer in real-time at 9600 baud, so special circuitry is not required here either.

Availability

The Rhino XR-1 is available for immediate delivery. For more information, write: Sandhu Machine Design Inc., 308 South State St., Champaign, Illinois 61820, or call (217) 352-8485.

The **Robotics VI Conference and Exposition** was held March 1 to 4 at Cobo Hall in Detroit, Michigan. For further information, contact the sponsor: Society of Manufacturing Engineers, One SME Dr., POB 930, Dearborn, MI 48128, (313) 271-1500.

For information on **Applied Machine Vision**, April 6 and 7 at the Sheraton Hopkins in Cleveland, Ohio, contact: Robotics International/SME, One SME Dr., POB 930, Dearborn, MI 48128, (313) 271-1500.

POWER FOR ROBOTS

David Smith
4505 Kennedy Blvd.
North Bergen, New Jersey 07047

A robot can use numerous types of storage batteries and power systems, and almost all of them outperform the common lead-acid storage battery. Yet the lead-acid battery is the most common storage battery, simply because of its price/performance ratio. Outside of a battery charger or power supply and a long extension cord, lead-acid car batteries are the cheapest source of power for a robot.

A car battery is heavy, filled with acid, vents hydrogen gas, and it has a long recharge time and a relatively low power density. It is designed to be kept at full charge, provide a short burst of power at a high amperage, and then be recharged immediately. Its power capacity is rapidly reduced at low temperatures.

A robot battery, on the other hand, should provide a low, continuous current over a long time, be capable of deep discharge without harm, be lightweight, capable of being stored discharged without harm, and contain no liquid to spill and damage any local electronic assemblies. Conclusion? The perfect robot battery is not a car battery!

Lead-acid batteries come in many sizes and amp-hour ratings. Sizes vary among batteries for motorcycles, garden tractors, snowmobiles, cars, trucks, golf carts, marine, industrial, power fail lights, and airplanes. Some aircraft batteries that are used in stunt planes will not spill acid when upside down. Lead-acid gel cells will not spill or vent much hydrogen gas.

I don't have much information about nickel-cadmium (NiCad) cells; however, any NiCad cell delivers 1.25 V, regardless of size. If you were to make a 12 V battery from individual NiCad cells, a 34 amp/hour 12 V battery would consist of 10 cells, weigh 40 pounds, cost about $60, and take up a space 9 by 3 by 14 inches.

I recall seeing a NiCad car battery advertised in a mail order car parts catalog. It might be cheaper than making a bank of smaller NiCad cells. I also remember an advertisement for a Tiny Tiger gas generator. It was not much

Lead-Acid Information

Time Needed to Recharge a Half Discharged Battery

Charger Amperage	15 Amp-Hour Battery	25 Amp-Hour Battery
7.5 amps	60 minutes	120 minutes
15 amps	30 minutes	60 minutes
30 amps	15 minutes	30 minutes
60 amps	15 minutes	15 minutes

Notice that increasing the amperage reduces charging time only to a point. Also note that 60 amps is a very high charging current.

Specific Gravity	State of Charge
1.26 to 1.28	100% charged
1.23 to 1.25	75% charged
1.20 to 1.22	50% charged
1.17 to 1.19	25% charged
1.14 to 1.16	Almost dead
1.11 to 1.13	Dead

Current Drain vs. Time

Load	Hours	Effective Power
5 amps	6.3	31.5 amp-hours
10 amps	3.1	31.5 amp-hours
15 amps	1.5	22.5 amp-hours
20 amps	1.0	20.0 amp-hours

Note that the lower current drain gives more effective power.

Power: Cranking Amperes-Amp Hours

Cranking Amperes	Reserve Capacity	Amp Hours
360	93 minutes	65
290	73 minutes	55
280	65 minutes	48

Typical Power to Weight Ratios

Lead-acid 37 watt-hours/kilogram
NiCad 33 watt-hours/kilogram

Sealed Lead-Acid Batteries

Temperature:

Charge −20 C to 50 C
Discharge −40 C to 50 C
Storage −40 C to 60 C

Life:

Standby 4 to 5 years
Cyclic use 250 to 1200 cycles

Conditions That Damage Batteries

Low electrolyte
Overcharging
Using tap water, not distilled
High discharge rates
Storing the battery uncharged

Power Available vs. Temperature

100%	80 Degrees Fahrenheit
65%	32 Degrees Fahrenheit
40%	0 Degrees Fahrenheit
18%	−20 Degrees Fahrenheit

larger than a car battery and would give an outdoor robot quite a long running time.

Golf cart and marine trolling batteries are designed for deep cycle discharge use. They cost more, however, and have a shorter life than regular batteries. They can be used harder, which means that your robot will run longer between charges, but I don't think that you will need this to develop your robot.

The life of a normal lead-acid car battery is inversely proportional to the depth of discharge. A battery that is normally discharged to one half of its rated amp-hours will have its life shortened to one quarter of normal life.

Car batteries range in price from $45 to $95 without trade ins. A trade in will get you about $5. If you don't want to get a new battery for your car and just keep the old one, you might consider buying a "junk" battery that still has lots of life left and can be purchased for $5 to $10.

The normal voltage for a standard 12 V automotive or marine electrical system with a generator is 14.4 V. If the system is operating correctly the system voltage should be in the range between 13.7 and 15.1 V. Charging voltage for a 12 V battery runs from 12 to 16 V. The normal open circuit voltage of a fully charged 12 V automobile battery is 12.4 V.

There is a lot of power in a standard car battery. Short circuits will melt heavy cable faster than you could believe! A typical heavy duty car battery has a reserve capacity of 97 minutes (while drawing a current of 25 amps). The average voltage will be 11.45 V times 25 amps — that's 286.25 watts each minute. So, 286.25 times 97 is 27,766 watts delivered to the robot. There are 746 watts per horse power, and a source of power that can deliver (746 watts times 60 minutes) 44,760 watts has one horsepower-hour. But our battery has only 27,766 watts, or 0.62 horsepower-hours. This means that our standard car battery will run a ½ horsepower motor for more than an hour.

Two Brevel motors moving a robot will draw 1.6 amps at 12 V — that's 19.2 watts. If those motors are turning 6-inch diameter wheels at the full load speed of 16 rpms, that's 300 inches per minute. Multiplying 5,280 feet per mile by 12 yields 63,360 inches per mile, divided by 300 inches per minute is 211.2 minutes to go a mile, or 3.52 hours per mile. The motors are pulling about 20 watts per minute. The battery has 27,766 watts on tap, which means the motors will run for 1388.3 minutes — that's 6.5 miles. With 6.5 miles as the radius of a circle, that works out to 20.6 square miles. If we cut that figure in half due to friction, hills, power drain from additional electronics, that is still 10 square miles that the robot could get lost in. Now you can see why I have designed limits as to how far my robot can travel.

Car batteries can blow up, and I don't mean a mild pop. Explosions usually occur when the battery is being charged, or when the electrolyte is low or frozen. Discharged batteries freeze easier than fully charged batteries. Don't charge a battery that is low on fluid or has ice in any of the cells. Charging or discharging batteries generates hydrogen gas, which

can explode under certain conditions. Make certain that the battery compartment is well ventilated. Hydrogen gas will not explode if the concentration is very low.

Remember to use distilled water to keep the plates covered at all times. Any battery should not be routinely recharged at more than 10 percent of its amp-hour rating, i.e., a 20 amp-hour battery should be charged at no more than 2 amps until a hydrometer shows full charge. A power cut-off timer for the battery charger is a good idea. A charging time of five to six hours is normal for a typical lead-acid battery. Before charging a battery, add distilled water to any cells that have a low fluid level.

The amp-hour rating is the amount of amp-hours that can be pulled from a battery over a period of 20 hours at 80 degrees Fahrenheit without going below 10.5 V. This is not as easy as it sounds. A car battery that is rated at 100 amp-hours is tested so that the amount of current that could be pulled from it was adjusted to last for 20 hours. This 100 amp-hour battery can deliver 5 amps at better than 10.5 V for 20 hours if the battery does not get any colder than 80 degrees Fahrenheit. Normal car batteries are tested at the 20 hour rate. The time rate parameter of the amp-hour rating is important because the capacity of the battery depends on the rate at which the battery is discharged. Slower discharge rates (i.e., reducing the average current drain) allow the battery to generate more electrical power than would be available at higher rates. The life (number of recharge cycles) will also be increased by lower discharge rates. If we were to discharge a 100 amp-hour battery at 16 amps instead of 5 amps we would find the battery dead after only five hours. This means that the battery produced only 80 amp-hours at the higher discharge rate of 16 amps.

Suggested Reading

A very good book on lead-acid batteries is *Secrets of Lead-acid Batteries* by T.J. Lindsay. The book can be ordered from Lindsay Publications, POB 12, Bradley, IL 60915. The company also has a number of similarly good books.

Other suggested reading includes *Lead Acid Batteries* by Hans Bode, copyright 1977 from Wiley-Interscience; *Storage Batteries; A General Treatise on the Physics and Chemistry of Secondary Batteries and Their Engineering Applications* by George W. Vinal, copyright 1955 from John Wiley & Sons. Interested readers should also take a look at the February 1968 issue of *Popular Electronics*, page 86, for information on connecting storage batteries in parallel.

Letters

Whither The Unicorn?

I would like to find out more about the Unicorn I Robot — where I can get information about what it is and how to build it? I first heard about it in the catalog *Robot Mart*, which shows a complete price list for parts but not much more.

Shiraz Ismail
114-25 Sunny Glenway
Don Mills, Ontario
CANADA M3X 1Z2

Unicorn I by James Gupton was described in a series of articles published during 1981 in Radio-Electronics *magazine. We checked with* Radio-Electronics' *publisher, Larry Steckler, to find an answer to your query: The entire series is available as a 52-page reprint book at a price of $12 plus $1 postage and handling. You can order it from Reprint Department,* Radio-Electronics, *200 Park Avenue South, New York, N.Y. 10003.*

Negative Feedback

Little error messages sent into our operational system of editing bring us closer to providing a product that reflects those most important qualities of *truth* and *accuracy*. We encourage readers to alert us to any errors that escape our editorial quality control process. Unless otherwise requested, we will give credit to the individuals who bring errors to our attention. The following technical error was found in an earlier issue and is corrected here:

In the November/December 1981 issue of *Robotics Age*, volume 3 number 6, there was a typographical error on page 13 in the article entitled, "Fast Trig Functions For Robot Control," by Carl F. Ruoff. A multiplication sign ("$\times$") was used in two places instead of a correct character. In the corrections which follow, we have underlined the *corrected* characters.

1. In the first paragraph of the left column, the following phrase is the corrected form:". . . then you can make use of the fact that

$$r = x \cos(\theta \underline{) + y} \sin(\theta)$$

to eliminate it.

2. The second equation, following the same paragraph, should read:

$$r = \sin\theta\,(x \cot\theta + y) = \sin\theta\,((x^2 / \underline{y) +} y),\ y \neq 0.$$

These corrections were phoned in by a reader who preferred to remain anonymous.

A COMPUTER CONTROLLED SENTRY ROBOT

A Homebuilt Project Report

H. R. Everett

329-B Ricketts Road

Monterey, California 93940

My robot's primary function is to randomly patrol a household while checking for fire, smoke, flooding, and intrusion, and take appropriate action if any of these conditions are found. It is appropriately equipped with numerous sensors, while still others are being planned. (Of highest priority is a tracking system that will allow the robot to follow a wire hidden beneath the carpet. This will make the repeated patrols a more orderly process.) To conserve battery power, patrols are made at random intervals. The robot spends most of its time in a passive intruder detection mode, where it actually has more available sensory inputs than when in motion.

A secondary purpose of this robot system is to serve as a mobile platform for research and experimentation in the areas of artificial intelligence, computer interface techniques, speech synthesis, and mechanical design. A SYM-1 single-board computer, mounted at mid-height on the front of the unit, forms the heart of the electronics. It functions primarily as a dedicated controller, but can be connected to a Synertek KTM-2 terminal through an RS-232C connection, thus greatly expanding the practicality of the overall setup. The robot remains motionless beside the terminal stand while the SYM-1 is connected to the KTM-2 terminal. The RS-232C connection provides both a power and data connection to the robot. When the robot is disconnected from the terminal, it proceeds under its own control on power supplied by a 12 V utility Die-Hard battery.

The mobile platform is equipped with numerous collision avoidance systems, including tri-directional sonar (forward, left, and right), four infrared proximity detectors, feelers, and, as a last resort, impact sensors. The data made available by these sensors is analyzed by the processor and the best course of action taken.

When not connected to the terminal, the robot has two modes of operation: passive and alert. In passive mode, the majority of sensors are activated but a good deal of the interface circuitry is powered down to conserve battery power. The robot relies on visual motion detection, ultrasonic motion detection, and hearing, to detect an intruder, while at the same time monitoring for vibration (earthquake), fire, smoke, toxic gas, flooding, etc. Some of these inputs are hardwired to cause an alert (switch from passive to alert status), whereas others must be evaluated first by software, which may trigger an alert if required. In the alert mode, all systems are activated and the robot is ready to respond to input information, or go on patrol. Either mode can be in effect while recharging, and recharging can be temporarily suspended if conditions warrant (what good is a watchdog that doesn't bark while he's eating?).

Recharging is handled automatically. The 12 V, 20 amp-hour battery provides approximately ten hours of service, and then requires about fourteen hours to recharge. Roughly one hour of power is available in which to locate the charging station (by means of a visual homing beacon) after the battery monitor circuits detect a low battery condition. The robot does not always wait for a low battery warning before returning to the recharging station. After a patrol, it may simply return to the recharger and wait until it's time to start another round.

The software is structured around a central loop that controls branching to various behavior pattern subroutines, which in turn call various library routines (such as sonar, speech output, time delay, steering, etc.) that can be used in many different routines. When the robot first starts from an off condition, each of the subsystems is automatically powered-up and tested. Voice output is used to announce the completion of each test.

To avoid the appearance of canned response to a given input, various signals are used to randomize behavior. For instance, the day of the week is represented by a 3-bit binary

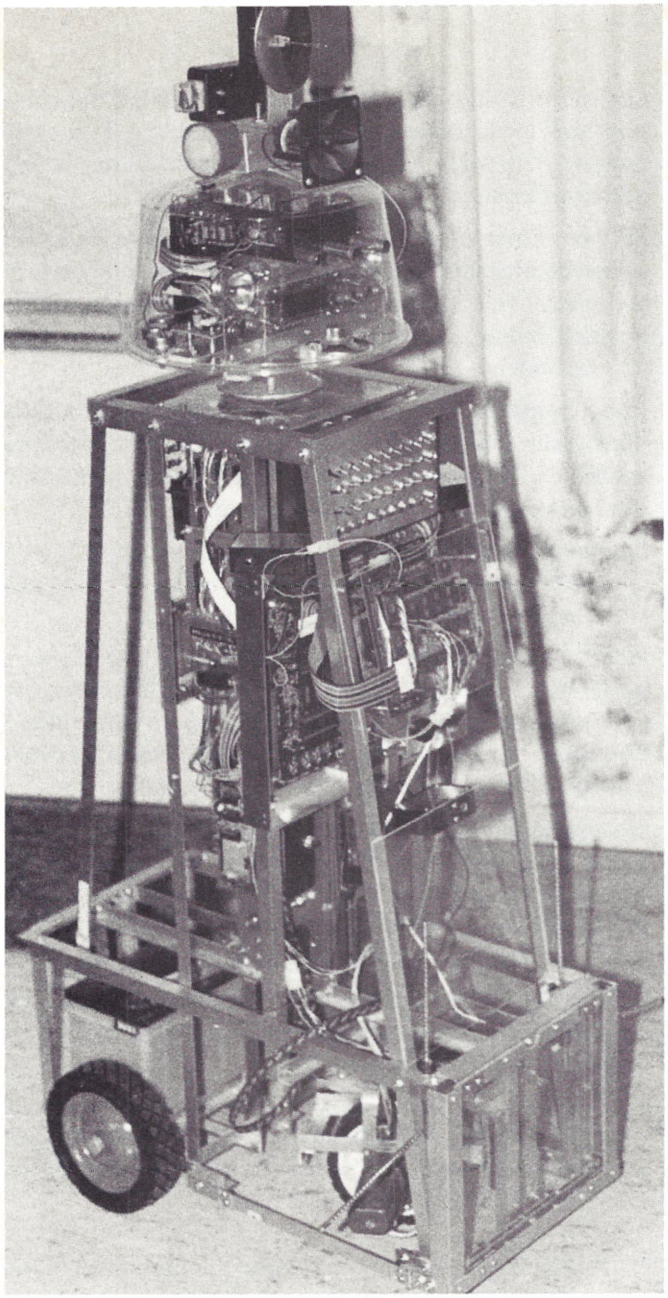


Photo 1: Overall View. Here is a full vertical view of the Robot assembly. The head sensors are for avoiding obstacles, animals or people. In this photo the newly added speech synthesizer is barely visible. A sonar transducer is mounted in front of the plexiglass panel. The ribbon cable wrapping around the right front frame member goes to a temporary EROM evaluation board. At present the Robot is a bare chassis without any formal attempt at making a finished, smooth housing.

number, and another bit tells the processor whether it is morning or night. By reading these bits before responding to an input, the response can vary according to conditions. A greeting of "Hello" in the morning could become "Hi" in the afternoon, for example. But this pattern is too easily detected. A more subtle variation would be to logically OR the AM/PM status bit with another piece of information, say the second bit of the day of the week, and use the result to decide which response to make. A random-number generator is also used to accomplish a similar result, both in speech synthesis and collision avoidance routines, giving the robot the appearance of having a mind of its own. The possibilities offered by software control of a system that is this complex make it an ideal platform for research and experiments in the field of artificial intelligence.

Processor Inputs

The robot uses many different sources of information to decide a course of action. Because I wanted to use this robot as an experimental platform, I provided a wide variety of sensors. The following list details those sensors that are currently connected to the robot, along with others that are still in development.

AM/PM: A single bit used to indicate morning or evening. It also drives a day-of-the-week counter.

Day of Week: A 3-bit binary code indicating the day.

Ambient Light: Signal for photocell on top of the head, which indicates if the room is light or dark.

Temperature: Two temperature sensing probes alert the processor if temperatures inside or outside the robot shell exceed or fall below predefined values.

Smoke: Photoelectric smoke detection circuit.

Toxic Gas: Figaro toxic gas sensor and detection circuit indicate presence of carbon monoxide, butane, propane, methane, etc.

Fire: Infrared fire detector acts as a back-up for the mechanical heat sensor.

Vibration: Seismic monitor indicates presence of vibration above a predefined value. Used for earthquake detection and detecting physical contact with an external object. This function is deactivated when the robot is moving.

Motion Detection: A National Semiconductor visual motion detector is used to detect intrusion. This function is deactivated when the robot is moving.

Peripheral Vision: The center photocell in the vision array feeds a comparator configured to respond to a sudden change in relative light input.

Optical Vision: A three-photocell array operates in conjunction with two small strobed spotlights in head. Performs two functions:

1. With spotlights turned on, the robot can locate and track reflections from light-colored objects.
2. With spotlights turned off, the robot can locate and track the beacon on top of the battery-charging station.

Infrared Scanner: Will be mounted on top of head, assists in initial location and identification of charging station beacon.

Flooding: Spring-loaded sensor indicates presence of water on floor.

Hearing: Directional hearing for intrusion detection triggers alert condition and initializes a left or right scan in the direction of disturbance, for visual, infrared, or sonic confirmation.

Head Bearing (relative): Analog-to-digital conversion, represents position of head by a 4-bit binary number (approximately 10 degree resolution), allowing the processor to know the direction in which the head is pointing.

Battery Level: Monitors battery voltage. Initiates recharging search subroutine when level falls below an adjustable set point.

Sonar: Can operate in many modes. Used in collision avoidance, indicates presence of sonar reflecting object within three feet of front sensor, then supplies range to nearest obstruction on left and right for processor evaluation. This system currently consists of one National LM 1812 transceiver with three fixed mount transducers (the left and right transducers are not yet in place).

The sonar can be used for intrusion detection within a range of ten feet forward, left and right by recording the range to nearest echo, and then triggering an alert if any range changes (robot must be stationary in this mode).

Infrared Collision Avoidance: Four fixed transmitter/receiver units indicate reflections up to four feet (forward, rear, left, right). One additional unit will be later mounted on the head, and can be positioned at any angle up to 110 degrees either side of centerline, for object detection, locating open doors, etc.

Contact Sensors: 16 momentary-contact switches are placed at strategic points and activated by front, rear, and side bumpers upon contact with an obstacle.

Feelers: Six 8-inch spring feelers sense nearby objects for collision avoidance.

Bus Status Monitors: Numerous comparators monitor the voltage on various power distribution buses and initiate shutdown procedures in the event of a malfunction.

Head Position Status: A 1-bit signal indicates when the drive wheel position matches the processor position command.

Barometric Pressure: A pressure transducer supplies digital input to circuitry which triggers an alarm if barometric pressure falls sufficiently fast to indicate approaching storm.

Storm Monitor: Lightning discharges detected by AM radio are rectified and fed to a capacitor whose voltage level is monitored by a comparator that sounds an alarm if discharges become frequent enough to charge the capacitor beyond a preset voltage.

Charging Status: Signals the presence of 12 V on charging probe. Once the 12 V is detected, the forward winding of the tandem drive motors is disabled, bringing the robot to a halt.

Overload (drive): Comparator monitors the voltage across the drive power circuit breaker. If a high drive current is detected, the computer assumes a drive wheel is stalled.

Optical Intensity: The intensity of light striking the center photocell in the optical array is converted to a 4-bit binary number. This value is used in intrusion detection and beacon search subroutines.

RF Data Link: An 8-bit input port is controlled by a radio transmitter located at the main computer terminal up to 90 feet away. This function is used to provide external commands if desired, overriding whatever program was in execution at the time.

Drive Direction: Tandem reduction motors drive a single 4-inch drive wheel. Two forward and one reverse speeds are available. The drive torque is sufficient to surmount small obstacles even with additional payload.

Steering: Driven wheel (front) can be positioned in any one of sixteen different positions for a maximum of 80 degrees left or right of centerline.

Head Angle: The head can be positioned in one of sixteen positions when under computer control, or can scan continuously back and forth up to 110 degrees either side of centerline, stopping at any point within this range when under control of the automatic scan and track system.

Floodlight: An omni-directional floodlight can be turned on to illuminate a room. A 555 monostable multivibrator triggered by processor command turns on the floodlight for 30 seconds, or until room lights come on. (This is used as an assist for household members returning to a darkened house.)

Spotlight: 4-inch reflector designed for use when intrusion is suspected, etc. Controlled directly by processor, via data distributor interface board.

Ultrasonic Energy Beam: 3-inch transducer mounted atop the head just under the parabolic infrared detector emits extremely high level ultrasonic transmissions (sound pressure levels in excess of 120 dB). It has two modes:

1. At a fixed frequency of 50 KHz, this device appears intolerable to insects. This mode is activated by the processor when the robot is recharging. The ultrasonic noise seems to do a good job of eliminating fleas and roaches, while not affecting animals or human beings.
2. The second mode is used to evict an intruder. The amplifier section driving the transducer is fed a sine wave that sweeps repeatedly through the range of 5KHz to 25KHz. This signal is in the audible region for a portion of this range, with sufficient intensity throughout the range to cause extreme discomfort, disorientation, nervousness, and possibly nausea. It is very directional in nature, and causes no permanent damage. But imagine the effect on an unsuspecting would-be thief!

Speech: The 280-word vocabulary is created using National Semiconductor's speech synthesiser DT1050 Digitalker. Appropriate speech subroutines are called by the processor in response to input conditions.

Transmitter: Used to activate beacon on top of charge station when battery conditions require recharging (also energizes charger).

Ultrasonic Control Link: Used to activate a BSR wireless remote-control system for controlling appliances using standard power lines (can control up to sixteen lights and appliances). This allows processor control of home environment. (This system is not yet fully operational and will see several modifications before its final configuration.) This link will eventually allow telephone input to the robot over the radio data link to be used to turn home appliances on or off in any predetermined or calculated sequence.

Alarms: A Sonalert driven by 555 timer circuits produces distinctive beep patterns for various alarm conditions, such as earthquake, fire, low battery, etc., each distinguishable from the other, followed by voice description of event triggering alarms.

Seven-segment Display: National Semiconductor's clock and temperature module alternately indicates the time of day and room temperature when robot is in alert status.

Bar Graph: 10-segment LED bar graph gives visual indication of battery voltage. An analog meter which can be switched among the various power buses (12 V, 9 V, 5 V, etc.) is also provided.

System Overview

The systems in this robot receive power via various distribution buses originating from the power supply board mounted on the lower left side of the vertical column. (Left and right will normally be associated with the front view of the unit, corresponding to the observer's own reference.) These distribution buses are watched by the monitor board, and routed to the appropriate circuitry through three multi-circuit branches. The power supply board provides 12 V, 9 V, 7 V to 9 V, 5 V, and 3 V. Provision is made to power down certain distribution lines when the robot is not in alert status. These include power to the 7-segment clock display, the optical board, the strobe board, the head drive motor, and certain of the interface boards. The drive relay.board, which also furnishes power to the infrared driver board, is powered down by software. Certain bus lines are energized only when the system is in a passive mode; these include the seismic monitor and peripheral vision (light level change detection).

Of prime consideration is conservation of battery power. Much of the original circuitry has been redesigned to make use of CMOS components versus the original TTL, and, wherever possible, circuitry is de-energized when not needed.The SYM-1 consumes approximately 1.2 amps, and now must run continuously, but the 32K-byte Beta Expansion Board can be powered down when not needed, as can the externally mounted read-only memory.

Just above the SYM-1 on the front is the input/output enable panel, which has numerous switches and push buttons for enabling or disabling certain inputs or outputs, as required by test sequences or operating conditions.

Below that, and in front of the SYM-1, is Beta Computer Device's 32K-byte programmable memory expansion board, and just in front of this board is the Netronic's speech synthesis board which utilizes the National Semiconductor Digitalker chip.

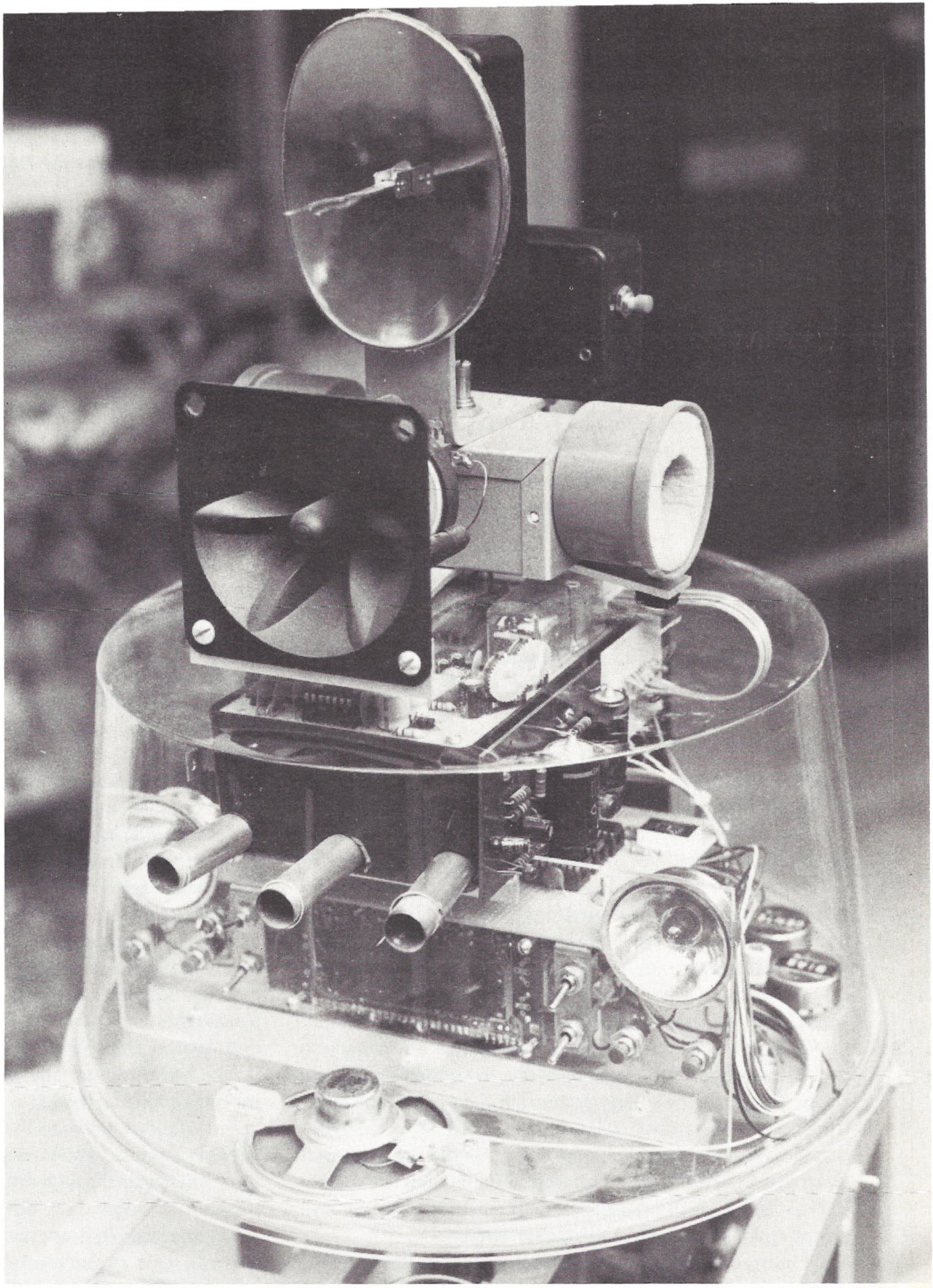
The drive wheel is powered by two 12 V actuators, mounted on either side of the wheel support cage. Each motor has a separate forward and reverse winding, with a common ground. An identical motor with a 2 ohm dropping resistor is used to position the drive wheel support cage (steering), and is coupled to a 10k ohm potentiometer which provides a 0 to 5 V analog signal to the analog-to-digital converter that supplies the interface circuitry with the steering angle expressed as a 4-bit binary number. A similar actuating motor is used to position the robot's plexiglass head, except that power is provided by a 7 to 9 V regulated supply, which allows the positioning speed to be adjusted to an optimum value for operation in conjunction with the fixed speed of the drive steering motor when the robot is in the tracking mode (i.e., homing in on the battery charger). The head motor and associated sensing potentiometer are mounted at the base of the robot, just forward of the battery, where there is more room, which helps keep the center of gravity low to the ground. The shaft extends up the center column, between the power supply board on the lower left and the sonar board on the lower right.

The monitor board is mounted on the power supply board, and it contains the circuitry for the battery level and power bus monitoring circuits, as well as the Sonalert driver circuits. The seismic monitor, which senses vibration, is enclosed in the plastic case that supports the monitor board, and its interface circuits are mounted on the monitor board, for convenience. The seismic monitor is reset by the robot going into an alert status, and triggering of the seismic monitor is wired to cause an alert. Since the robot is, by design, in an alert status when in motion, the seismic monitor is automatically deactivated unless the platform is immobile.

The various interface boards are mounted above the monitor board on the left side. These provide the necessary comunications between the processor and the real world it seeks to control. Also incorporated is expansion to an additional 32 inputs through data selectors, and an additional 32 outputs through data distributors. These are all driven simultaneously in a unique fashion by a common four-line address, which is also used to provide a head positioning command to the head controlling circuitry. (If I/O lines were scarce, it could also provide the steering command in a similar fashion, but since the SYM-1 is equipped with 71 I/O lines, and 64 more were added, the steering interface was given its own I/O port.)

The rear sensor panel holds the smoke detector, resurrected from an "as is" sale, the mechanical backup fire detector, the Sonalert, and the meter that monitors the various power distribution buses. Two banana plugs supply power to a logic probe when troubleshooting, and the temporary "evaluation" hookups, which the platform is designed to sup-

Photo 2: Closeup Of The Head Assembly. This illustration shows some of the details of sensors. The parabolic reflector will be used with an infrared detector to help locate the feeding station beacon. A black ultrasonic transducer and power amplifier is located below the reflector. Directly under the plexiglass cover is the strobe lamp board and its photocell array of 3 pickup tubes. Below this are boards containing circuitry for a clock/calendar integrated circuit and a thermometer sensor. Voice output is provided by a speaker.



port. On the lower right of the rear panel is located the AA Simulator Board. When the SYM-1 is disconnected from the actual sensors and connected to this board, commands can be fed to the interface boards in real-time for circuitry evaluation and troubleshooting. This is invaluable for stepping through a complicated sequence to eliminate bugs.

The sonar board is located on the right side of the robot, and is built around National Semiconductor's LM 1812 Sonar Transceiver. The circuitry provides a maximum range of about ten feet (twenty feet round trip). When operating in the collision avoidance mode this range is decreased through software to about two and a half feet by simply limiting the time spent waiting and watching for indication of an echo. This allows us to reduce range without having to alter transmission power. The sonar can also operate in a motion detection mode at full range when the platform is immobile, by simply calling up the Sonar Intrusion Detection subroutine.

The battery charging system is mounted in a small box midway up the charging tower. The homing beacon that allows the robot to locate the tower is at the top of the tower. At present this beacon consists of a 75 watt light bulb, positioned at the same height above the floor as the optical photocell array in the robot's head. Future plans call for the installation of a pulsed LED source to emit infrared light that would be detected by the parabolic infrared reflector mounted on top of the head. The charger's positive output is fed to an inverted pizza pan that serves as the base of the tower, and also the point of contact for the robot's charging pickup probe. The ground is connected to the metal pole that forms the beacon tower, and is electrically isolated from the pizza pan by a plexiglass insulator (see photo).

The charger pickup probe is mounted on a small box attached to the drive wheel support cage, and it is always positioned directly in front of the drive wheel. This probe routes

the + 12 V from the charger through a diode on the monitor board to the battery via a fuse on the power supply board. The circuit is completed when the spring-loaded front bumper makes contact with the charging station pole. The processor then reads the "charging current present" input and stops the drive wheel. As a backup measure, a sensing relay in the charge probe circuit simultaneously disables the forward motor windings. In more than 200 observed dockings, this arrangement has only failed once — when a piece of rug lint snagged on the pickup probe. The lint insulated the probe from the pizza pan as the robot closed on the charging tower, and the processor interpreted the bumper contact with the pole as the striking of an obstacle because the front impact sensor triggered an interrupt and the probe status did not indicate contact with 12 V. The robot responded by backing up to the left, verifying a lock on to the beacon, and approaching from a slightly different angle, this time with success. The pickup probe was later modified by soldering a smooth contact to the probe end (previously a flexible spring).

The robot is brought into alignment with the charging tower through a complicated but reliable process that consists of locating the light source, verifying that it is the beacon on the charger, tracking it with head angle while turning towards it, and then running into the charger tower to make an electrical contact. (The verification step was added as an afterthought: in an earlier case the automatic scan and track circuit accidentally locked on to a table lamp situated at the same height as the optical photocell array in the head which caused the robot to attempt docking with the lamp.)

When the battery condition goes below the set point for more than five seconds, the flip-flop on the monitor board changes state and the processor initiates the recharging search subroutine. The transmitter is activated, which turns on the battery charger and homing beacon at the charging station. The processor also enables the automatic scan and track circuit on the optical board. The head begins to scan left and right, seeking a point source of light of sufficient intensity to trigger the center photocell comparator. This scan action is controlled by the scan flip-flop on the optical board in the robot's head. The scan flip-flop is reset and set by limit switches at both extremities of head travel, causing the motor to reverse direction and the head to scan the other way. This action continues as long as no light source is detected. If any of the three optical comparators goes high, indicating a light has been found, the scan flip-flop is deactivated and the tracking inputs take over control of the head positioning motor.

The tracking inputs come from the left and right photocells in the array. Their respective comparator outputs indicate a greater intensity either side of center, in reference to the center photocell output. The appropriate drive winding is energized, and the head turns to regain maximum intensity at the center photocell, thus tracking the source. (If by chance both left and right photocells showed intensities greater than center, the head remains motionless.) When the array outputs indicate the head is correctly positioned (i.e., pointing at the source), the head positioning motor will be de-energized.

Once the processor ascertains that the head has located the beacon, it interrogates the source to verify that it is indeed the beacon. Verification is accomplished by turning off the transmitter and watching to see if the beacon also goes out. If the source is not the beacon, the scan is reinitiated and the process repeats.

Assuming the source is the beacon, the processor reads the head position and sends a command to the steering motor to position the drive wheel at the same angle. As the platform turns, the head automatically tracks the source, and the bearing to the beacon decreases. The processor subsequently decreases the turn angle until eventually the source is directly in front of the platform.

The task of docking with the recharger is greatly simplified by the design of the charging tower, as well as the pick-up probe. The charging tower can be approached from any direction. The geometry is such that as long as the front bumper contacts the upright pole, the probe will be touching the pizza pan underneath. Because the front bumper is 12 inches wide, there is a very reasonable margin for alignment error, so even if the robot is traveling so as to miss contact with the tower by, say, one inch, a last-minute correction command from the processor (at a point as late as six inches from contact) would still allow plenty of time for a perfect docking. (At that close range, the head would be indicating a 45 degree bearing to the beacon, instead of the desired zero degrees. A subsequent 45 degree turn in the desired direction would result in contact close to the center of the front bumper.)

The optical photocell array can also operate in a second mode, in conjunction with the two small spotlights mounted on either side of the clock display in the head. In this mode, the automatic scan and track circuit seeks out and tracks reflections from light-colored objects. While scanning, the spots are strobed by the strobe board mounted on top of the box that supports the three photocell tubes in the head. This conserves power; the processor turns the spotlights on continuously as long as there is a reflection. The spotlights produce identical areas of high illumination on either side of the center reference photocell. These are detected by the left and right photocells on reflective surfaces, up to a range of about five feet. Beyond this point, the beams converge and there is no relative brightness with respect to the center photocell, so objects beyond five feet are ignored. This system is extremely limited in that only light-colored objects can be detected, and it can be swamped by ambient light. (It was incorporated because it provided justification for the flashing eyes, which all robots are expected to have.) It does, however, produce some interesting responses if a passerby in a light-colored shirt arouses the robot's attention.

Conclusion

The design and construction of a robot is a continuous project, for whenever one system is perfected, another problem or another approach to a problem arises. It's a project that will teach anyone about hardware and software solutions.

Android Design: Practical Approaches for Robot Builders

by Martin Bradley Weinstein
Published by Hayden Book Co., Inc., Rochelle Park, New Jersey, 1981
ISBN 0-8104-5192-1, 248 pages softcover, $11.95
Reviewed by Paul Hollingshead

With a title as promising as *Android Design*, who could resist running down to the book store for a copy? Not me! The reward is a fun-to-read description of the author's approach to creating an android. The book does not present detailed plans for one fixed design, but discusses the requirements and some solutions for many of the features you might want an android to have.

Early in the book, the author tackles the sticky problem of defining what the term "android" means. As used in this book, it is "A mobile mechanism capable of manipulating objects external to itself under the constant control of its own resident intelligence . . ." The goal he has set for his own machine is a 4½-foot-tall mechanism that moves on a pair of tracks. A cylindrical trunk can bend forward and back, and rotate about its axis. Near the top of the trunk are two arms. The hands at the end of the arms can grab objects or push a button. On top of the trunk is a head, jam-packed with electronics: two cameras, two microphones for speech recognition, and a speech synthesizer.

A major design consideration is the problem of moving around an indoor environment with dozens of obstacles. Weinstein presents a plan for avoiding damage to the android and to the other objects and creatures in the rooms.

The basis for the design of the author's drive mechanisms is the ability to climb and descend stairs, and to fit in hallways and through doorways. The author discusses the geometry of the android's base in detail. Specific sizes are given for both a tracked drive and a triangular wheel drive, and the motors they require.

To hold the various objects the android may want to handle, Weinstein has come up with an ingenious hand with fingers made out of chain. Cables on either side of the chain can be used to curl or straighten the fingers. For a firm grip, three fingers are opposed by two thumbs. One finger can be operated independently of the others, to push buttons, for example. Weinstein suggests several different transducers — including temperature — to give the fingers a primitive sense of feel. Protecting the whole assembly requires nothing fancier than rubber gloves.

The cleverest item in the book is the "ramera," a digital camera based on a dynamic memory chip (an MK4008), with its cover removed. An image is formed by writing 1s to all positions, then reading the memory to see which bits have been changed to 0s by the light falling on them. Gray scales can be determined by doing a series of reads from the memory.

Collision avoidance for the android can be provided by several dozen optical and ultrasonic detectors. The author explains where the sensors should be placed, and why. They could be used in a real-time mode to detect objects that are moving, and also to create a simple map of fixed objects for future use. A binary code is suggested to describe the objects' location and to weigh the probability of whether or not they're still there.

To control the android's functions and interpret all the data requires a brain. The author suggests a network of microprocessors, organized in layers. Each layer improves the information it receives and passes it to the next higher layer. Simple organizational charts are given for five major subsystems. Several pages are devoted to describing a number of microprocessors that could be used.

A brief look at voice perception and response is included in the latter part of the book. A number of off-the-shelf systems for these tasks are mentioned. The author gives possible vocabularies for recognition and answering by the android. An appendix with a list of references is included.

The best feature of the book is the large number of addresses given for part suppliers. Motors, gears, transducers, IC's, batteries, and relays all have suggested sources. Articles explaining how to apply several of these parts are also mentioned. The wealth of information provided here will give you an excellent.

ROBO/PLOT III

TRS-80 Mod I or III Full Control Interface.

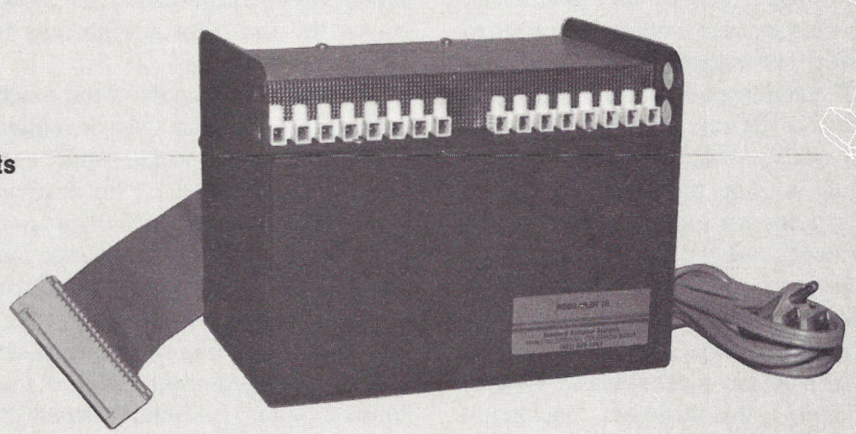
- 8 inputs
- 3 stepper motor outputs.
- 2 reed switch outputs.
- On board power supply.

- Develop your own low cost CAD/CAM systems.
- Control robotic functions via basic and machine Language.
- Build plotters using third stepper for multi-color wheel or other mechanical operations
- Build P.C. board drilling machine, odd shape glass cutting machine, sandblast art, soldering fixture and many more.
- Joy stick input to digitize your programs.
- Sufficient power on board to operate 12 volt North American Philips K82954-M3, stepper motor, or smaller.

Total Cost
$249.50 P.P.
with manual, disk or tape demo programs

SEND:
CK—MO
VISA
MCGH.

Attn:
- Schools
- Industry
- Hobbyists



- Custom programming in basic, assembly and fortran.
- Stepper motors-lead screws-belt drives, etc. available.
- Custom manufacture of components or systems.
- Other interfaces forthcoming.
- Your comments and ideas WELCOME.

ALL THIS AND YOU CAN STILL USE YOUR COMPUTER TO DO YOUR BOOKS.

SHIPPING! STOCK TO 6 WKS.

Stanford Software Systems
Box 226 — Manitou Springs, CO 80829
(303) 685-1013

CIRCLE 15

NATURAL LANGUAGE UNDERSTANDING

A First Look

Bil Lewis

325 Central

Menlo Park, California 94025

The basic problem in dealing with computers is communication: how to tell the machine what to do. There was a time when all human to computer communication was achieved by programming in assembly language the machine-level instructions that the machine hardware processed. Programming in assembly language is time-consuming and arduous, but in the early days when human time was cheap relative to the cost of the machine, it was a reasonable thing to do.

As machines became cheaper and programmer's time more expensive, other languages were devised that required less work from the person, and more from the machine. Logically, this process will continue until *no* work is required of the programmer. In other words, instead of programming, the programmer will simply *tell* the machine what is desired, and the machine will figure out how to do it.

This concept of simply telling a computer what to do in English (or French or German), rather than programming it, is the essence of *natural language understanding.* (The term *natural* language refers to any language which people commonly speak: English, Swahili, or Esperanto. The contrast is with *formal* languages, such as first order calculus or Pascal.)

As can be seen from figure 1, the ability of a machine to converse in a natural language vastly simplifies the job of the programmer (for all examples I will restrict myself to English as representative of natural languages). In this example shown, the program is eliminated entirely, allowing the person who desires the information (but doesn't know the programming language) to access it directly.

Understanding Natural Language

In some sense a computer inherently "understands" its assembly code. When it comes across the code for "JUMP TO LOCATION 0100", the machine knows that it must set its program counter to 0100. In the example shown, our machine must first "understand" the question to figure out which country has the fastest submarine. This is a formidable problem to which we have no satisfactory solution — indeed, we have no more than the vaguest notion of what understanding means.

Happily, for our purposes, it is not necessary for the computer to understand the question completely in order to answer it. If the machine is able to construct a representation of the question which contains sufficient information for it to construct a database query like the one in figure 1, then we might claim that it "understands" the question well enough for all practical purposes. That this representation totally ignores the fact that submarines travel under water is unimportant.

In restricting the scope of our machine's understanding to only that information which is required for the desired computation, we have vastly simplified the problem and have eliminated the need for an enormous amount of "world knowledge." To further reduce complexity, it is also possible to focus the natural language understanding system on a single subject, or "domain of discourse."

The LADDER system, designed at SRI International in the late 1970s, is a natural-language-access-to-navy-database system that represents the state of the art in natural language understanding. LADDER restricts itself to information contained in a database about ships, and it is unable to handle questions concerning world knowledge beyond the contents of its database (e.g., "Why does a carrier need an escort?") or questions about any other domain of discourse ("To what country does the fastest airplane belong?").

Within these limitations we can decide upon some sort of formal representation for the knowledge of interest and for the sentences that the ultimate user may wish to type. The network shown in figure 2 is fairly typical of those used in most natural language systems. Few carry any more information than is shown there, though the form may differ enormously. The main problem faced by natural language understanding at the present is not that of devising more complete knowledge representations (although that time will certainly

come), but rather that of reliably getting from the language to the chosen representation.

Natural Language Understanding

Computational linguistics attempts to quantify language and to decide mechanically what is and is not a part of the language. Its goal is to build a grammar (i.e., a set of rules) that will determine whether or not a given sentence is a proper, meaningful element of the language.

Contrary to the claims of most high school English teachers, there is no complete grammar for the language. Moreover, there isn't even a consensus about what the language comprises. Figure 3 provides good illustration of the language we call English.

There exists a central core of sentences that most people accept as "proper." (I'm using the definition of sentence to be "A string of words terminated with a period.") Outside of this core, there is a set of sentences that some people will accept and others won't, though typically they can understand them. Finally there is a category of possible "sentences" that almost no one will accept.

While there is an interesting problem of deciding how best to define a language and what is grammatical it is not germane to the problem of building natural language systems. Designers of language understanding systems are most interested in having a grammar that is broad enough to understand most sentences that users are likely to give it, and narrow enough to be computationally feasible.

To define the grammar of a language, most computational linguists use a set of production rules that are typified by those shown in figure 4. These rules tell the machine how each element of the language can be expanded and combined to produce a phrase structure grammar (PSG).

Essentially, a phrase structure grammar consists of one "start symbol" (here, <SENTENCE>), an arbitrary number of nonterminal symbols (such as <NOUN PHRASE> and <DETERMINER>), and a set of terminal symbols commonly called "words." A legal sentence is any that can be obtained by starting from SENTENCE, following the production rules, and ending with a string of words (i.e., no nonterminal symbols).

```
English query: "To what country does the fastest submarine belong?"

Datalanguage Query:

BEGIN
DECLARE Y1 STRING (,100) ,D = "]"
DECLARE Y2 STRING (,100) ,D = "]" Y2 = "00.0"
DECLARE Y3 INTEGER Y3 = 0
DECLARE Y5 STRING (,100) ,D = "]" Y5 = 0
DECLARE Y4 STRING (,100) ,D = "]" Y4 = 0
FOR R1 IN SHIPCLASCHAR WITH (R1.TYPE2 EQ "S") AND (R1.TYPE1 EQ "S")
  FOR R2 IN SHIPCLASDIR WITH (R2.SHIPCLAS EQ R1.SHIPCLAS)
    FOR R3 IN SHIP WITH (R3.UICVCN EQ R2.UICVCN)
        BEGIN
        Y1 = R3.MCSF
          IF Y1 LE "99.9" AND Y2 LT Y1 THEN
            BEGIN
            Y2 = Y1
            Y5 = R3.NAT
            Y4 = R3.NAM
            Y3 = 1
            END
              END
IF Y3 EQ 1 THEN
  BEGIN
  NSTDPORT.STRING1 = Y4
  NSTDPORT.STRING2 = Y5
  END
END
```

Figure 1: Comparison of a question asked in English and the same question asked in a hypothetical database language. While a human can recall information stored in memory without knowing details of how they were stored, a computer (usually) must know how the information is represented in memory.

When we wish to "understand" a sentence, we use a program called a parser to build a parse tree like that shown in figure 5. Once the parser has produced a tree that indicates the syntax (i.e., the structure) of the sentence, we can begin a semantic analysis of the given structure to decide its meaning.

In the case of language understanding programs, the meaning that we are going to try to extract will be in the form of some formal representation as mentioned above. The formal representation that the program produces will (hopefully) encode exactly those aspects of the sentence that are pertinent to the domain of discourse and the desired response.

It should be noted that we are claiming no "psychological reality" for either the way that the sentence is parsed or the formal representation that is produced. There are some researchers in the field who claim that their methods of processing language mimic the thought processes of the human mind, but I feel there is scant evidence for this.

Although the ability to produce a formal representation of a sentence is still a long way from having the machine respond in the desired fashion, it is a clear step forward from the enormous problems associated with natural language. In particular, formal representations avoid the problems of ambiguity and context dependency, which have plagued natural language systems from the beginning.

The first of these problems, ambiguity, arises when a word or sentence has numerous meanings. For example, when I say "bank," do I mean the edge of a river, or a financial institution? Entire sentences are often ambiguous, as shown by a comparison of figures 5 and 6.

Machines must handle ambiguous sentences in much the same way that people handle them. Either they ask for clarification ("Did you mean that the dog had the stick?"), or they figure it out from the surrounding context ("The boy broke a branch from the tree. The boy hit the dog with a stick.")

Context dependency, however, cannot be used to solve the ambiguity problem because our programs are not up to that level of sophistication. We cannot expect a machine to "keep in mind" the conversation and remember that something I say now depends upon something that I said before.

Most programs written to date are restricted to considering one sentence at a time, with no reference to what has gone before. Thus, the computer would give the same response to identical questions, even though it would be inappropriate (see figure 7).

Because of our present inability to handle these and other problems, any system we build must not be expected to solve them. Moreover, users must either be aware of the limitations, or be informed of them at times when they might occur.

Present Applications

Up to this point I have stressed what natural language understanding systems *cannot* do. I did so to deflate the numerous unrealistic claims that were made in the early days of computer science. The truth of the matter is that we are only now beginning to understand the *complexity* of the problem. But lacking the correct solution to the problem does not prevent us from performing research, nor from producing systems of great practical value. To begin with, let us review the main restrictions, and then consider how possible applications match up against them.

a. The natural language system will not have any general knowledge of the outside world.

b. It must operate in a single domain.

c. It cannot remember the context of the "discussion."

d. It must be able to handle ambiguous input in some reasonable manner.

One of the first projects for a natural language system that comes to mind is that of machine translation, an area where it was anticipated that computers would prove immediately valuable. But if we consider our small set of restrictions, we see that a translation program would be faced with solving both a and c.

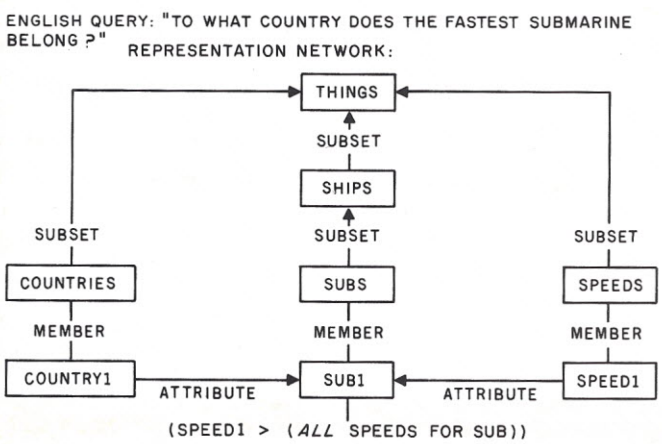


Figure 2: Contrast of an English sentence with a network that attempts to represent its meaning.

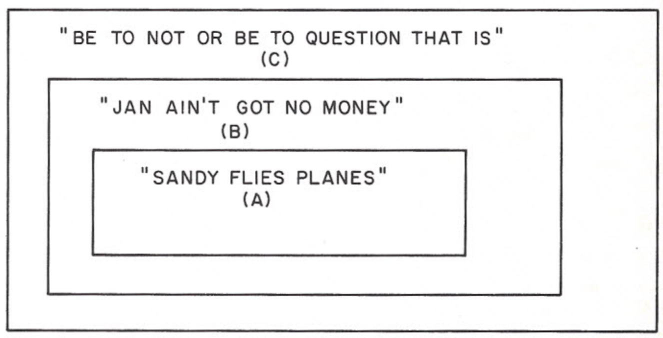


Figure 3: In the English language (as in any language) there are: (a) a certain set of sentences that are syntactically correct and acceptable to almost everyone; (b) an encompassing set of sentences that are acceptable to some and understandable by most; and (c) a larger set of sentences that are nonsense to almost everyone.

< SENTENCE >	= =>	< NOUN.PHRASE > < VERB.PHRASE >
< VERB.PHRASE >	= =>	< VERB > (< NOUN.PHRASE >) (< PREP.PHRASE>)
< NOUN.PHRASE >	= =>	(< DETERMINER>) < NOUN > (< PREP.PHRASE>)
< PREP.PHRASE >	= =>	< PREPOSITION > < NOUN.PHRASE >
< VERB >	= =>	hit | talk | swim etc.
< NOUN >	= =>	boy | dog | stick
< DETERMINER >	= =>	the | a | an
< PREPOSITION >	= =>	with | of | in | on etc.

Figure 4: An extremely simplified set of grammatical rules for understanding the English language.

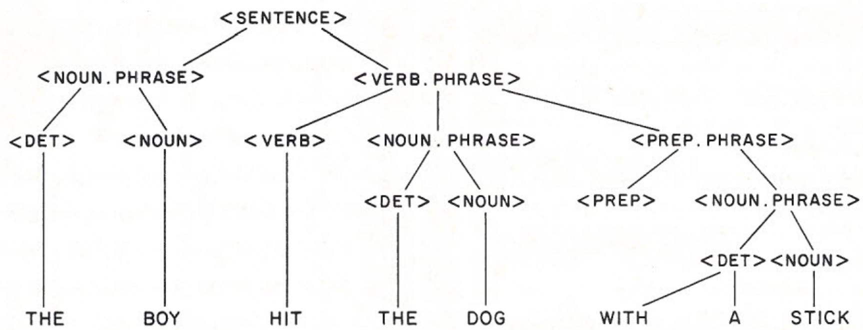


Figure 5: A parse tree for the sentence, "The boy hit the dog with a stick." This interpretation tells us that a boy has taken a stick and hit a dog with it.

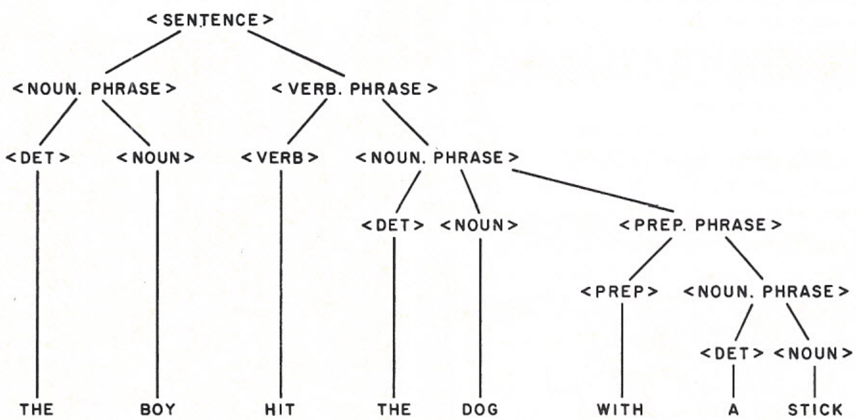


Figure 6: Another parse tree for the sentence, "The boy hit the dog with a stick." This interpretation tells us that a boy has hit a dog which is holding a stick. Notice how the stick has moved from the possession of the boy in figure 5 to the possession of the dog in figure 6.

A prime example of the kind of translation we can expect from a program is illustrated in figure 8. Over the past few years there has been some creative work in providing machine assistance to translators, but the goal of an independent machine doing complete translation of text remains unrealized.

A second area of application, and one that has seen the greatest success, is that of database access. When working with a database, there is seldom any reason for the machine to need knowledge outside of the database in question (although we can run into the type of problem illustrated in figure 7). The domain of discourse is strictly limited to the contents of the database, and rarely do questions require the context set by previous questions.

Database access systems also have the advantage that a very definite, correct response is expected, and there is a clearly defined method of obtaining it. The method is, of course, the translation of the question into a database query (as shown in figure 1), and the response is the result of that query.

The combination of these attributes has made database access the one area of application that has moved into the nonacademic world. There are a small number of natural language access database systems, such as LADDER, that

READER SERVICE

Product#	Company
1	Ackerman Digital Systems, Inc., p. 5
2	AMSI Group, p. 43
3	Doring Associates, p. 9
4	E/Z Associates, p. 15
5	General Numeric, p. 2
6	Intellimac, cover IV
7	Mack Corp, p. 1
8	Micromint, p 43
10	Robot Mart, p. 15
11	Robotics Directory, p. 20
12	Sandhu Machines, cover II
13	Sorrento Valley Associates, p. 18
14	Space Studies Institute, p. 17
15	Stanford Software, p. 29
16	Westinghouse, p. 7
17	Ypsilanti Area Idc, p. 1
18	LogiGate, p. 20

are used by such agencies as the Navy and NASA, and even one, INTELLECT, that is commercially available and in use at about a dozen sites.

Another area of application that appears ripe for exploitation is that of machine control. Until now, all control protocol and programming languages for robot arms have been very formal and well removed from the realm of natural language. If we consider the limitations on natural language systems, we see that the types of things we would like to tell our robots to do fit well within these constraints.

At present, if a new task is to be designed for a typical arm and vision module, programming abilities are required. While this is not an unreasonable request in a research environment, it can be asking too much of a factory or business location. For this reason, we can soon (5 to 10 years from now) expect to see robots being controlled by factory personnel "talking" to them from terminals.

Conclusion

In the long run I anticipate many of the wonderful accomplishments that have been predicted — a typewriter that takes dictation, a translation program for Russian novels, and a program that will write all your programs for you. I'm afraid, however, that the year 2100 is a better estimate than 2000. Therefore, let me concentrate on those types of natural language systems we can hope to see in the next ten or twenty years.

We can expect to see "more of the same" in database access. There will be natural language database access systems on small machines that computer-naive users will be able to set up and run.

For machine translation of texts we'll find a "computer assistant" that will give rough translations, sentence by sentence, for inspection and correction by a human translator. It will be able to help with spelling and some grammar, and it will have an on-line dictionary for consultation, but will be incapable of independent translation.

In medicine, geology, and other professional fields involving exploration and extrapolation from large amounts of statistical data there will be computer consultants that will give advice and provide statistical analysis of data. Once again these programs will not operate independently, and their advice will have to be carefully considered, as they will unquestionably make occasional false (and potentially fatal) assumptions.

The one fact that must be emphasized to all who use these programs is that they will not "understand" what they are doing. Thus, there is no danger of machines replacing human beings, but they have the potential to eliminate the dull, mindless tasks that humans have performed for centuries.

First interaction; only one question.

User:	"List the managers and their locations"	
Program:	"John Smith	Philadelphia
	Joe Cool	San Francisco
	Jane Doe	Boston . . ."

Second interaction; two questions.

User:	"How many stores do we have on the east coast?"	
Program:	"Seven"	
User:	"List the managers and their locations"	
Program:	"John Smith	Philadelphia
	Joe Cool	San Francisco
	Jane Doe	Boston . . ."

Figure 7: As shown here, computer database systems do not remember parts of conversation that have occurred in the past. The first interaction correctly produces a list of all managers and their locations. The second interaction should produce a list of managers in stores on the east coast. This would not be difficult for a human to understand but the computer sees these questions as unrelated.

The Yankee's batter hit a fly ball.	The New Englander's mixture of flour and liquid struck a tight collection of small insects.

English => Russian => English

Figure 8: An example of typical problems in current attempts to translate one language into another that is totally under human control.

RT-13 VIDEO/SOUND RECOGNITION SYSTEM

Stephen Montero
5015 Glickmin Street
Temple City, California 91780

RT-13, thirteenth in a series of robots built at the California State Polytechnic University, is an excellent example of the application of multi-processing techniques. Although functionally simple (a manipulator has not yet been constructed for this unit), the robot uses complex software imaging techniques to recognize and track various household furnishings. RT-13 can also locate a position described through voice commands after charting an xy two-dimensional map of the respective area. The primary tasks for this robot are object recognition and avoidance. The knowledge gained from the construction and operation of RT-13 will be used in the creation of the more complex RT-14 currently being built.

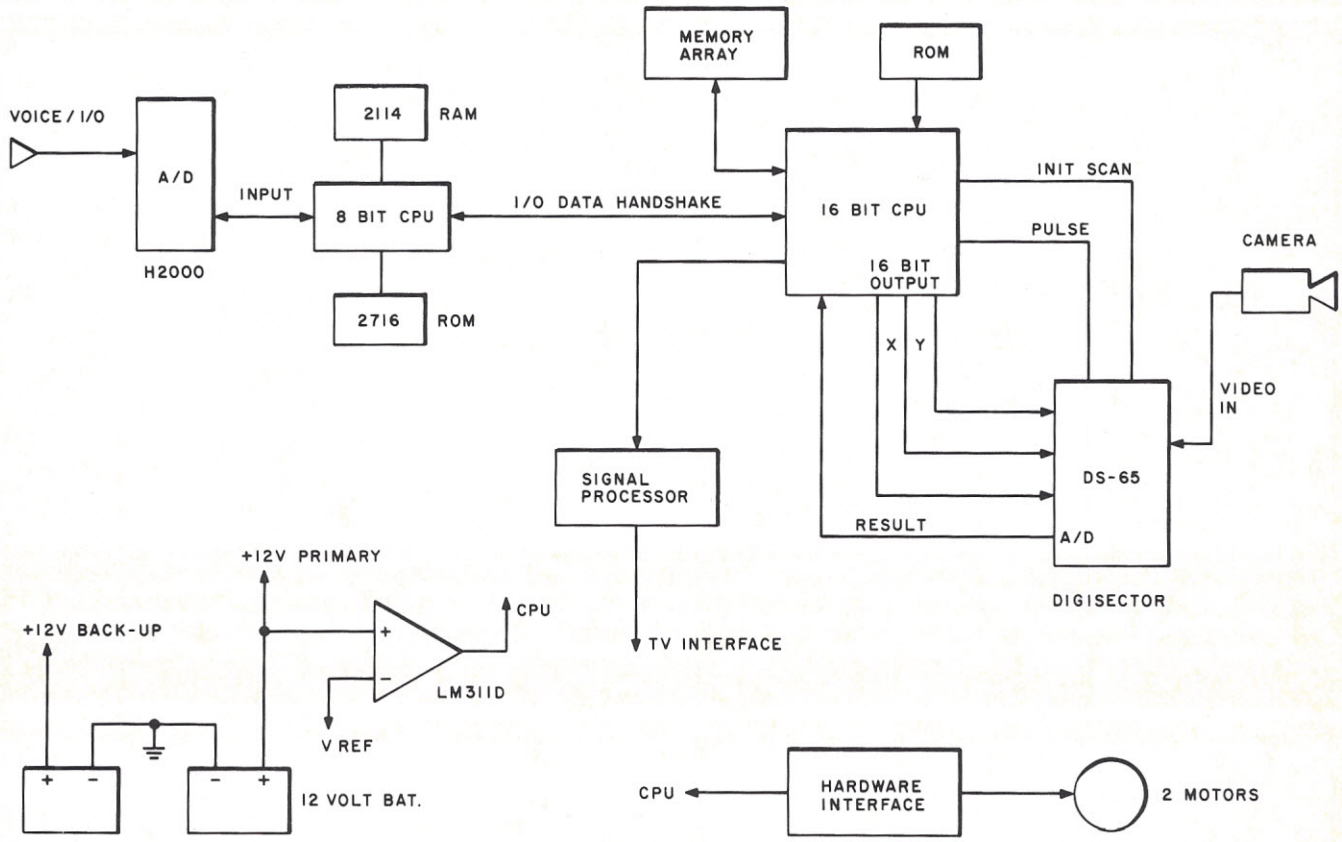


Figure 1. RT-13's onboard control system. This block diagram shows the overall organization of the present RT-13 vision and voice response computing system. The goal of this design is implementation of search algorithms to acquire an internal model of its room. RT-13 can then respond to appropriate voice on commands specifying a location.

Most of the software was developed on a PDP-11, RT-11 system and later downloaded into the robot's on-board system. For voice recognition, RT-13 uses a 6502 based 8-bit computer linked to a Heuristic's H2000 speech recognition system. Major direction decisions are also made through this computer in accordance with feedback from the 16-bit video processor.

The body was built from common materials such as lexan, wood, and aluminum. ■

RT-13 placed second in our Home Robot Photo Contest. Steve Montero received a check for $50 and a one year Robotics Age *subscription extension.*

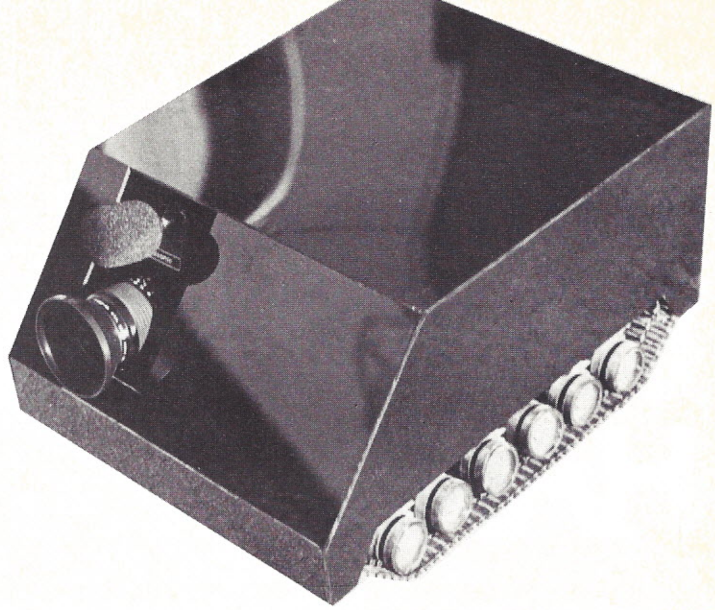


Photo 1. RT-13 compares what it sees to pictures stored in memory of tables, chairs, etc. It is designed to avoid objects found in its path.

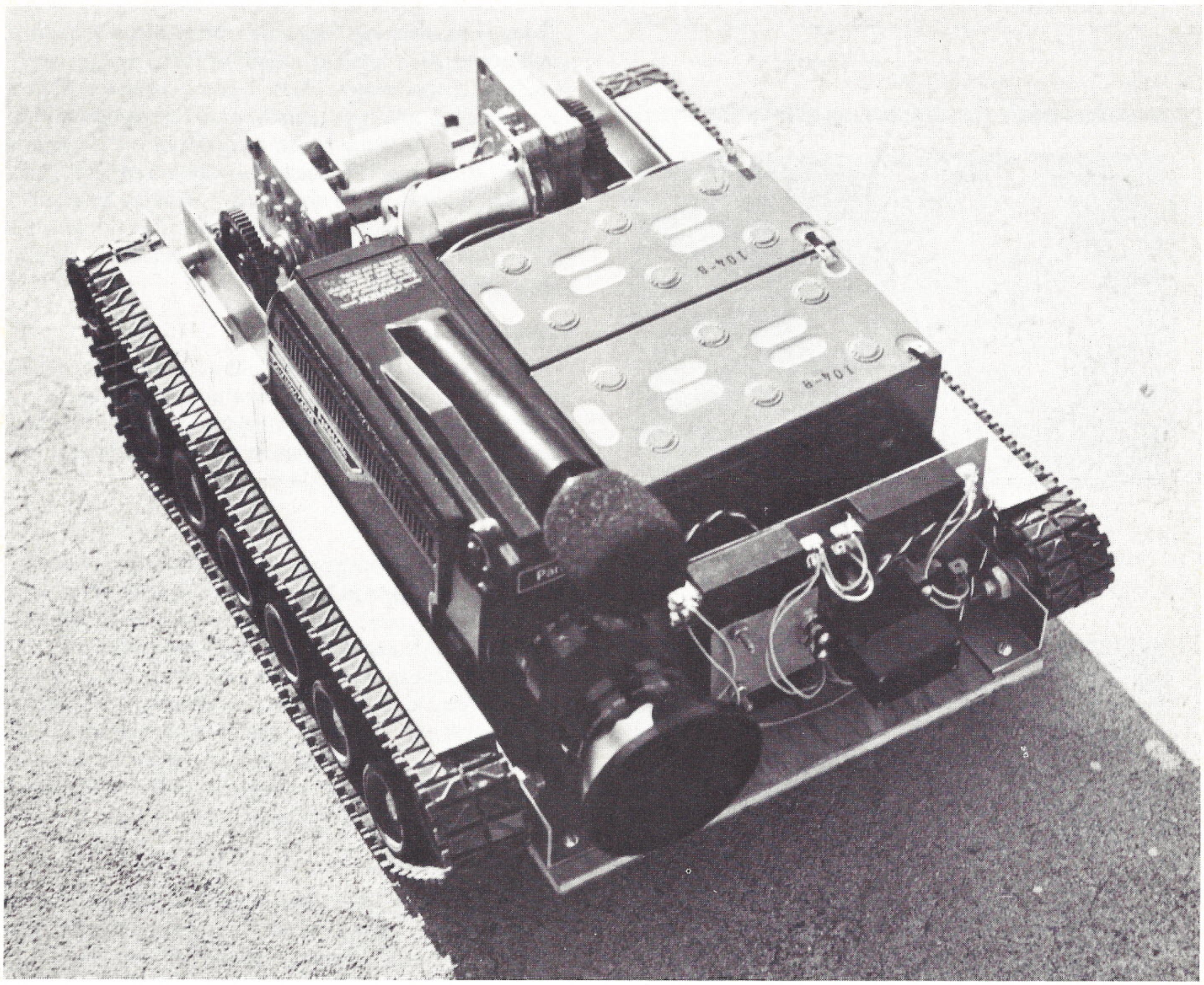


Photo 2. A glance at RT-13's interior. Although some hardware has been removed for this picture, you can see the video camera with built-in microphone, primary and backup 12 V gell cell batteries, and the gear arrangement for driving the treads. The home-built touch is evident from the wooden base.

AN INEXPENSIVE HAND

Mark J. Robillard
3 Peach Lane
Townsend, Massachusetts 01469

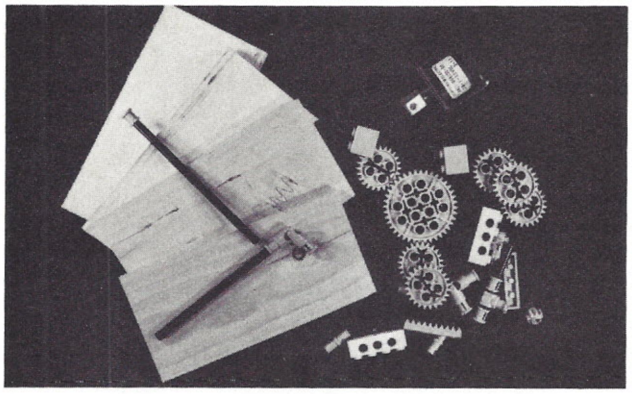


Photo 1: A working robot hand can be built from these parts. Gears and miscellaneous parts appearing on the right are from the Lego kit.

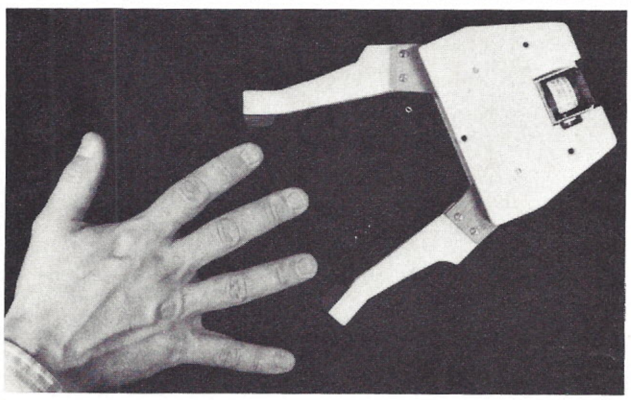


Photo 2: Completed hand, shown in comparison to human hand. Small ball bearings, solenoid, and linear actuator can be seen.

If you're an electronic engineer or technician, the first thing you discover when entering the field of hobby robotics is your own lack of mechanical skills. I come from the field of microprocessor systems where the most mechanical component you ever see is the plastic card ejector on the edge of a printed circuit card. Typically there are no pulleys, gears, or cables, whereas there are as many or more mechanical components to a robot as there are electronic control circuits. That means gears, pulleys, and cables. So what's an electronic mind to do?

Experiment

Kits are always a good start, but for the price you pay for some of those experimental arms you could buy several TRS-80 Color Computers. Not that the prices aren't justified; mechanical parts, let alone stepper motors, aren't cheap. But if you can't afford a kit, what do you do?

Do It Yourself

Start with one of those bargain bags of motors and a handful of gears, pulleys, and cables. Where from? Well, there are a number of sources, and I've included a list of the more popular ones in the textbox that accompanies this article. Of course, if you've ever sent for or seen some of the catalogs that are available, you know that words like pitch, tangent, etc., describe a basic one-inch gear with the same amount of teeth as the two-inch gear you picked out on the preceding page. The only reason you could pick out the two-inch gear was that there happened to be a picture. Help!

Enter the Toy Store

Stop everything! Drive, bike, or walk to the nearest toy or department store. Stroll over to the section where they sell building kits for kids. Lego . . . remember that word, for it holds the key to solving the problems of the weekend robot experimenter.

You want gears? They've got 'em. They're heavy grey plastic instead of the metal variety, they all have the same

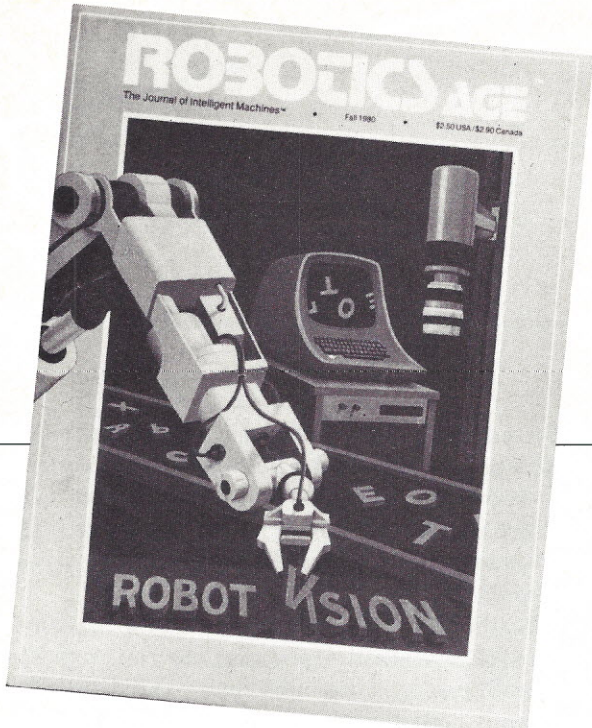


Photo 3: Cover of Fall 1980 Robotics Age*, which provided the inspiration for the robot hand design.*

Robot experiments don't always need large amounts of money. Functional systems can be built from spare parts and children's toys.

pitch so the gear teeth mesh together, and there are four basic sizes. Photo 1 shows some parts from a kit with four $^3/_4$-inch gears, one $1^1/_2$-inch gear, four or five $^1/_4$-inch gears, two linear and one 1-inch differential gear that looks like a gear with teeth that bend over the edge. In addition to the gears, you get various shaft sizes, a universal joint, and miscellaneous parts. All for less than $8! You probably won't want to rely on these parts for your final robot, but it makes mechanical experimenting as easy as picking up a battery for your radio.

Construction

There are hundreds of ways to approach the construction of a basic hand. Should it have five fingers, or just the usual two-pincer style? Will it close totally on one command, or will it have the ability to grip objects of varying shapes and sizes? All these questions have to be answered.

The easiest design, and the one described here, is the two-pincer open-closed version. Even if you are interested in more sophisticated designs, this is a good place to start. While building this hand, you will gain mechanical experience, and using it will provide valuable hands-on training in the world of robot manipulators.

Structure

The first thing you see when you look at a robot hand (photo 2) is the structure that holds the mechanical parts together. It is usually made of some sort of metal that is bent using a tool called a brake and cut with a shear. Don't have either tool? The word to remember now is *wood*. Did you know that prominent industrial designers use wood to build prototype mock-ups of the most popular minicomputers and controllers, and that many of the ads you see in magazines contain a picture of the mock-up? It is almost impossible to tell that they're not metal. For hobby use there's no reason to rely solely on metal parts (unless you are lifting a rather large weight, and at this point that should not be a factor).

Wood has many advantages. First, it is easily cut, machined, or drilled. In addition, all this can be done by hand. Try cutting $^1/_8$-inch aluminum with a nibbling tool for more than five minutes and you will quickly appreciate this aspect. For experimental purposes you can't go wrong, and the cost is lower too, not to mention the availability.

The same toy store you found the gears in will probably also contain small sheets of Pactra brand modeling plywood. These plywood sheets come in various sizes and thicknesses.

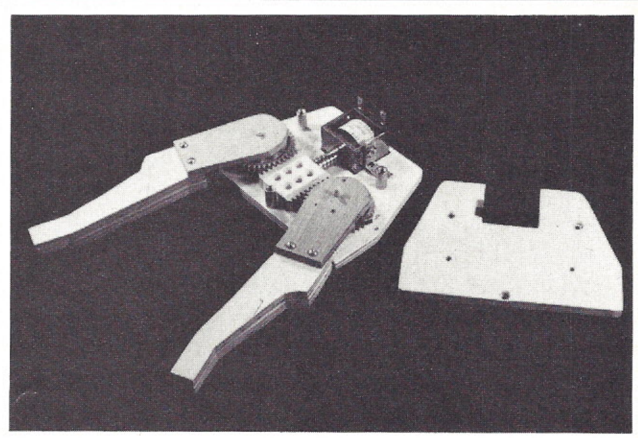


Photo 4: Inner workings of hand are shown with the top cover removed. Note that the action of linear actuator pulls the fingers together.

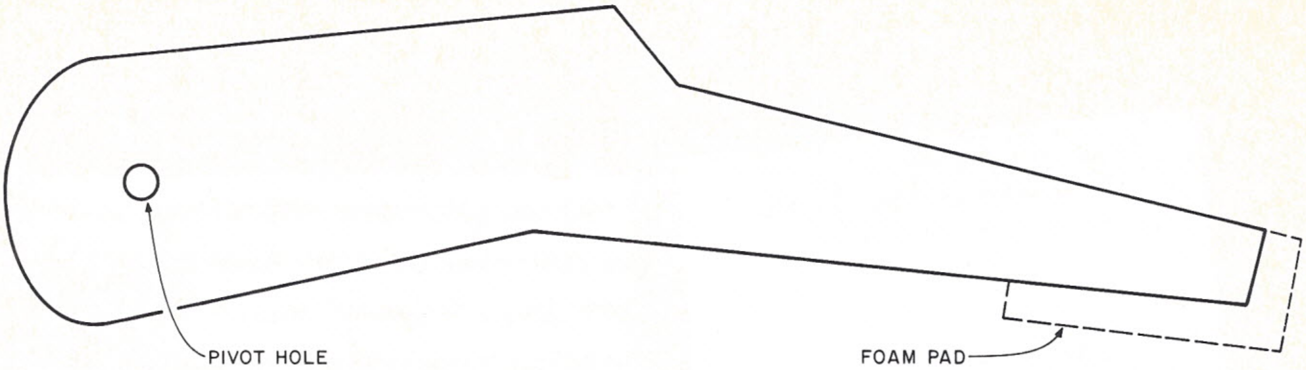


Figure 1: This actual-size finger pattern can be used to trace over the material being used for construction. The pivot hole should match the size of the pivot shaft being used.

They typically cost about $4 each, providing a reliable and easily worked material for this kind of project.

Now, pick a shape. The cover of the Fall 1980 *Robotics Age* magazine (photo 3) shows an artist's rendition of a very stylish mechanical arm with an equally interesting hand-manipulator connected to it. It looks useful enough. Why don't we build our hand as close to that one as possible?

Fingers

Since we have decided our hand will have pincer-style fingers, there are only two digits to worry about, and they are the same size and shape. The first step is to cut two identical fingers-pincers out of wood or whatever material you feel most comfortable with. Next drill the hole that will be used for the pivot point. I used an $\frac{1}{8}$-inch hole because my pivot shaft was an $\frac{1}{8}$-inch dowel. Figure 1 is an actual-size pattern for the finger.

After the pincers are finished we'll take our first step into the field of mechanical engineering. How do we get the fingers to open? Looking at your own hand you'll see that the fingers must move apart in opposite directions an equal amount. Open the gear kit, take out the large $1\frac{1}{2}$-inch gears, and place them on the pincer parts so the center hole is directly over the pivot hole drilled in the finger. I glued the gear and finger together at this point. You can fasten them however you please, but the finger and the gear must be one-and-the-same.

Now we can look at different ways of moving the pincers. Figure 2 describes two approaches. We could put a motor between the pincers, with a gear (smaller) that meshes with the two finger gears. The result, however, is that one finger moves in one direction and the other follows in the same way. This does not close the hand. A still smaller gear could be placed

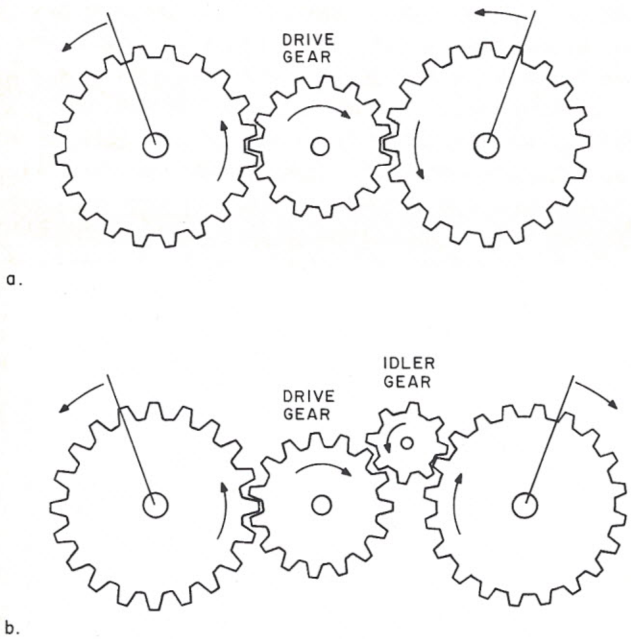


Figure 2: Original gear idea (figure 2a) moves the two finger gears in the same direction at the same time. The addition of an idler gear (figure 2b) reverses rotation of one finger gear and moves the fingers in opposite directions.

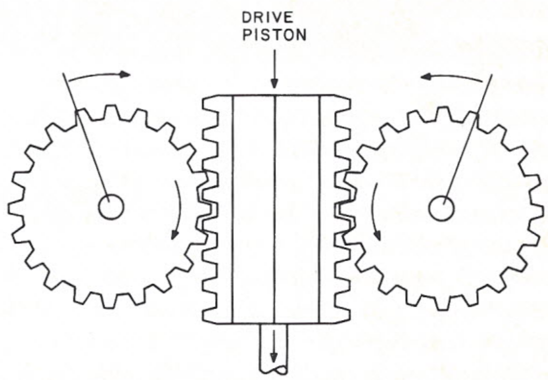


Figure 3: A different approach to finger control uses linear gears. No idler gear or motor is required. Solenoid action pulls the drive piston down, which turns the finger gears inward. A spring returns the piston to open position.

between one finger and the motor. Yes — this will reverse the direction of movement of the finger. We've got it! Now let's look at the motor requirements.

Motors

Assume the fingers are being held apart by a spring. This is necessary to help the hand return to the open position. Pulling them toward each other expands the spring. This takes a certain amount of force. I picked a motor out of the bag, set up the gear mechanism in the base structure, and attempted to energize the motor. Every motor in the bag stalled! I even tried changing the spring size and tension until the spring didn't hold the fingers open at all. Nothing worked. It became obvious that, in this case, bargain bag motors were not for hands.

Earlier in the week I had purchased a handful of solenoids at a local surplus store. All that had come with the solenoids was the coil and the shaft; no spring to hold the shaft up so that the energizing action of the coil could pull it down. In my junk box I found a spring that just happened to fit over the shaft. The solenoid was exerting an enormous amount of power pulling this shaft down. How could I use this device?

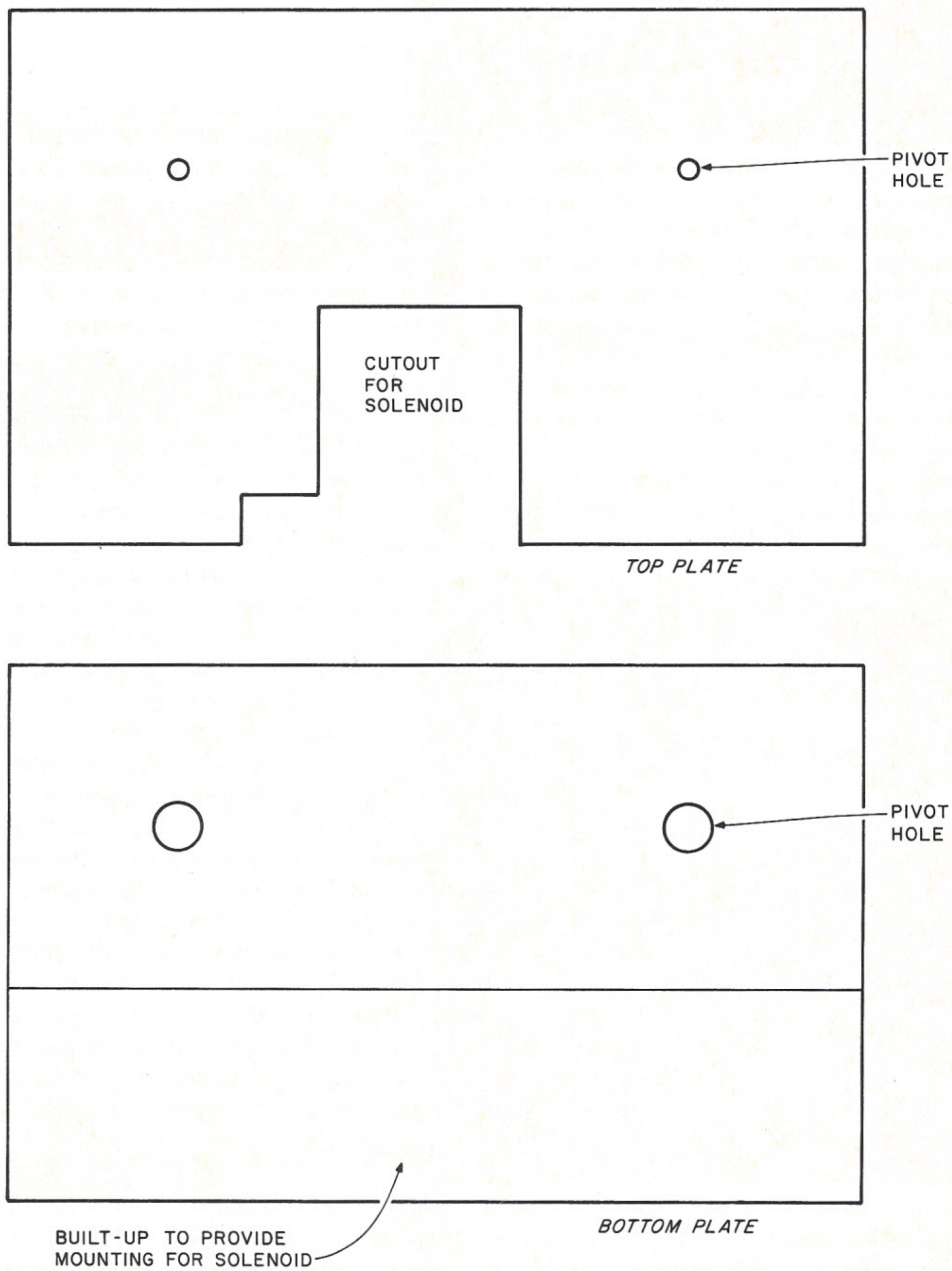


Figure 4: Actual-size patterns of top and bottom plates used to secure gear and finger assembly together. These patterns are simplifications of the plates used in the model depicted in the photos.

Lego . . . remember that word, for it holds the key to solving the problems of the weekend robot experimenter.

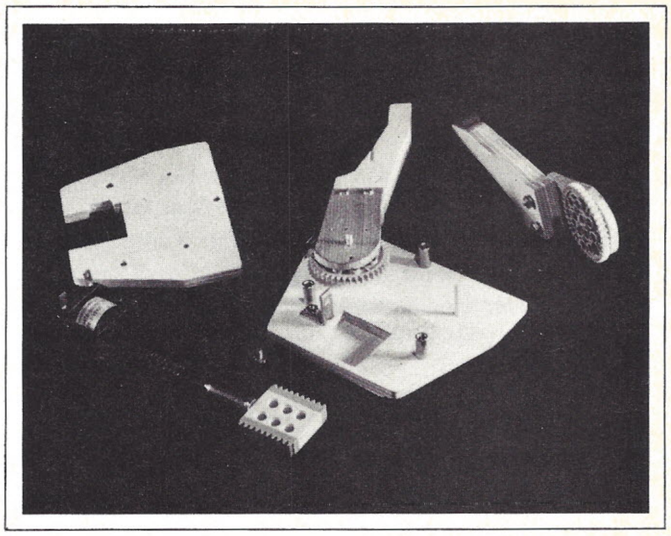


Photo 5: Disassembled hand, showing individual parts and construction details of finger-gear attachment.

Enter the linear gear found in the Lego kit. This gear looks like a flat piece of plastic with teeth on one side. Figure 3 shows the linear gear arrangement. If you place one linear gear so that it meshes with one of the finger gears, and then pull the linear gear down, the finger will move from left to right. Let's mesh the other linear gear with the other finger and pull together. It works!

Now we must space the two linear gears apart so they can mesh with the two fingers. I used several miscellaneous kit parts, drilled a hole in one end, and glued the solenoid shaft to the assembled block. Then I slipped the spring over the shaft and placed the shaft into the solenoid.

Now for the big moment — turn on the solenoid. Clack! The linear gear block smacks into the coil housing. Place the two finger assemblies in to mesh with it this time. Clack! The two fingers come together (and probably fly apart somewhere in the room). The next structure will hold the entire assembly together.

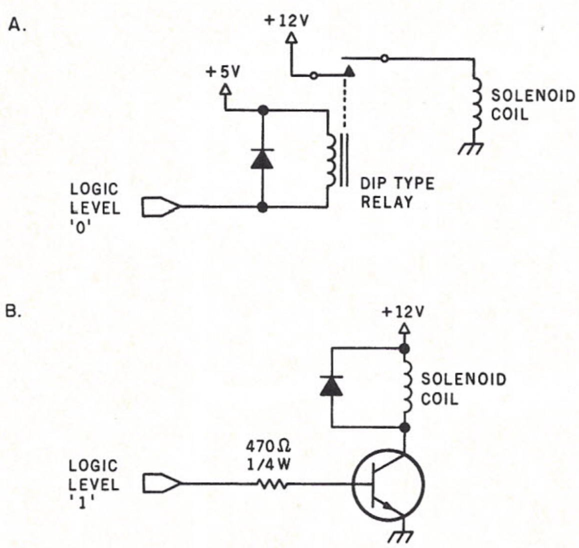


Figure 5: Two ways to control the finger solenoid. Figure 5a shows a relay control circuit utilizing a TTL control relay. A logic level zero (approximately 0 V) closes the fingers. Figure 5b uses a transistor circuit to control the fingers. The transistor can be any NPN power transistor. A logic level one (approximately +5 V) will activate the fingers.

Back to Structures

Pull out the wood again. Figure 4 illustrates the structure plates. Measure the distance between the two finger pivot holes and mark them on a piece of board, or simply place the fingers in the desired position and mark through the holes. Then drill holes to accept the shafts that will be put through the pivot holes. At this point I drilled slightly larger holes and inserted small ($^1/_4$-inch) ball bearings into the holes. This allows the shaft to move easily. If you can get ball bearings for this purpose, do so. (The sources listed in the textbox should be able to supply you with many different-sized bearings.)

Do the same for the other board or plate on the opposite side of the hand. Study the photographs for more detail on the plates and how they interconnect to the full mechanical system. In this case, pictures are far better than words, and I've tried to show as many views as possible. If at this point the assembly doesn't move smoothly, you'll have trouble when it comes to energizing the solenoid.

The actual control of a single solenoid is very easy and not very interesting. There are two basic ways to accomplish it: first is the old traditional relay control; second is the more modern transistor control. We will explore both methods because each has its merits.

Relay Control

We will assume that eventually some sort of semi-intelligent controller will be commanding your robot. This assumption alludes to a logic level type of control system ($1 \geq 3$ V, $0 \leq 0.8$ V). If this is the case, then a relay sensitive enough to respond or energize with these voltage levels will be used (see

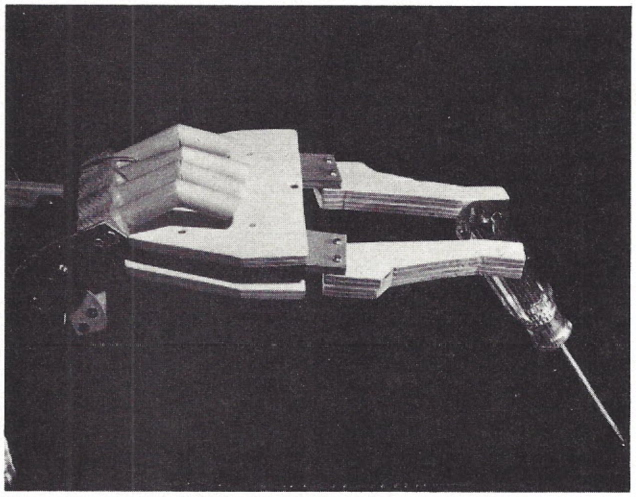


Photo 6: Hand in action, gripping screwdriver. Note the use of foam pads on ends to increase holding action. Note use of a second generation manipulator in this pose . . .

figure 5a). Basically, the relay in the circuit supplies power to energize the solenoid when the relay is turned on by a logic zero. Of course, a logic one could be used if connected through an inverter. Simple isn't it? You could probably build the circuit on a very small board and mount it right on the solenoid.

Transistor Control

The approach shown in figure 5b is just as simple. The figure shows a logic level one activated version. All you do is use the transistor as a power relay. This circuit is equally as small, but it is no longer necessary to mount any control circuit. You may elect to have the circuit that commands the wrist provide the control for the hand.

That brings us to the next step, the future.

Where Do We Go From Here

That's up to you! I'm going up the wrist to discover stepper motors and those gears with the teeth that wrap around the side. I'll be getting into microprocessor control and plastic tubing. If you're still interested, please join me on the yellow brick road of beginning robotics.

Mechanical Parts Sources

Hobby Robotics Company
Box 997
Lilburn, GA 30247

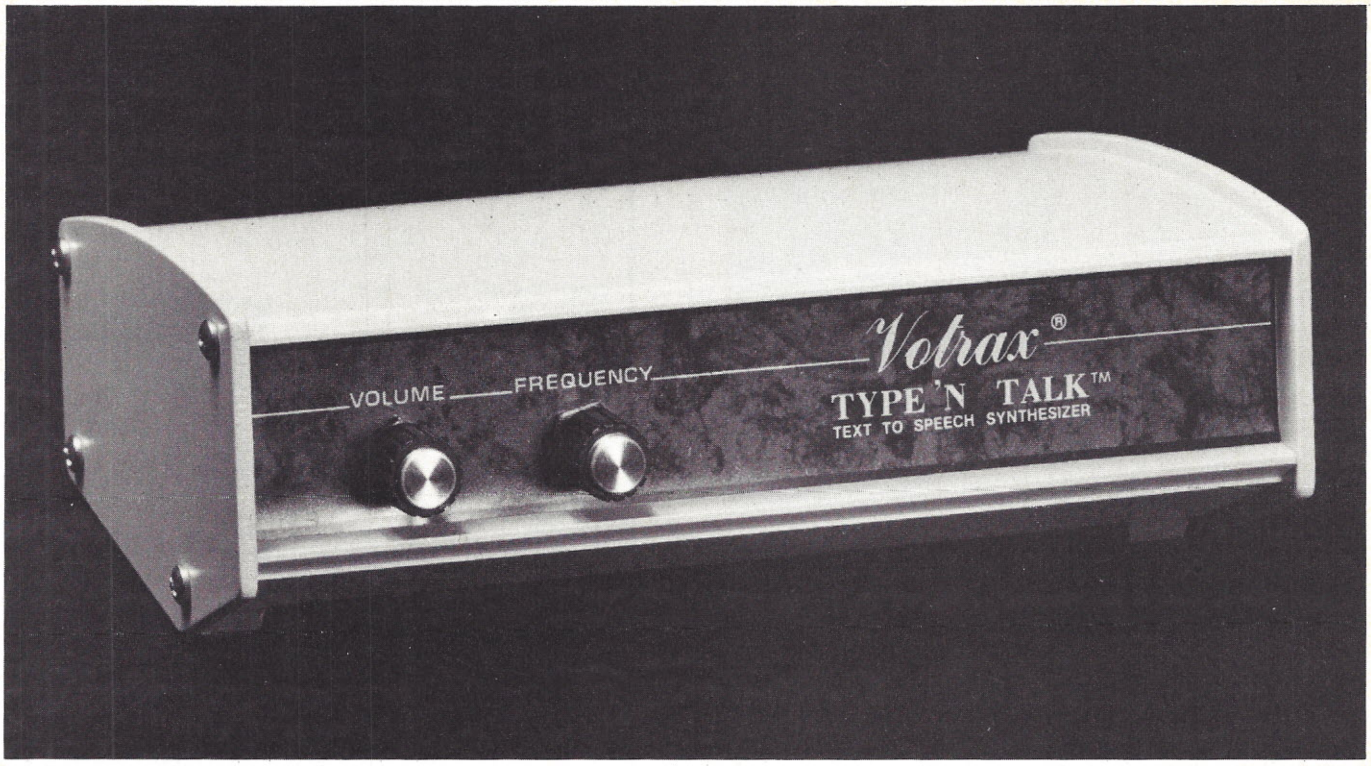
Robot MART
19 West 34th St.
New York, NY 10001

Winfred M. Berg Inc.
499 Ocean Ave.
Rockaway, NY 11518

Edmund Scientific Company
101 East Gloucester Pike
Barrington, NJ 08007

Herbach and Rademan Inc.
401 East Erie Ave.
Philadelphia, PA 19134

Photo 1: The Type-'N-Talk speech synthesizer.



TYPE 'N TALK

Paul Hollingshead
1235 Wildwood Avenue, #30
Sunnyvale, California 94086

Unlimited vocabulary speech synthesis for personal computers has arrived in the form of a new device called Type-'N-Talk (a trademark of Votrax). For roughly $400, Type-'N-Talk can provide voice output for any computer or terminal that has an RS-232C serial interface, thus adding a new dimension to your system's abilities.

An ASCII text string (up to 749 characters at once) followed by a carriage return is all it takes to produce reasonable quality verbal output. The Type-'N-Talk unit has its own microprocessor which converts the text into a collection of basic sounds. Any of these 64 different basic sounds, called phonemes, are then converted to audio form, using the company's SC-01 custom chip. An on-board 1 W amplifier boosts this output to drive an external speaker.

A series of phonemes can be used instead of normal ASCII text to produce special effects or words that are difficult to pronounce. The descriptive manual includes a list of example words, the phonemes used to construct them, and the ASCII codes that represent the phonemes. This information is enough to get you going on your first experiments with the device.

Isolated words occasionally sound less than perfect. After all, creating speech is a tricky task. However, there is no problem understanding what is being said when a complete sentence is spoken, and strange sounding words can be adjusted either by careful, deliberate misspelling, or mixing phonemes with normal text.

In addition to unlimited vocabulary and a common RS-232C interface, Type-'N-Talk has several other handy features. The first unit can be connected to other Type-'N-Talks or to a printer. Each unit can then be individually selected. The Type-'N-Talk can also be connected directly to a modem.

Several control codes can adjust the operation. Under program control, the unit can echo the phonemes as they are generated, if you want to see or record them. The Type-'N-Talk can be instructed to ignore the data sent on the serial line and pass it on to the next unit with which it is connected. If a carriage return is not sent with the data to be spoken, the unit will wait slightly more than four seconds and then speak. A two-character control code can disable this feature.

One of the slickest features is the way upper- and lowercase letters are handled. Normally both are treated the same.

An alternate mode doesn't do anything special to a single capital letter at the beginning of a word. But if two or more uppercase letters are placed together, they are spoken individually. A sentence output as "The 74LS136N chip is faulty" would be pronounced as "The seven four ell ess one three six en chip . . . ." This option can be used to pronounce abbreviations in text.

As received from the factory, the Type-'N-Talk package consists of a thorough 32-page manual, a baseball-sized power supply, and the main box. The primary unit is about 8 by 5 by 3 inches (20 by 13 by 7 cm) and weighs two pounds (900 g). Adjustment knobs for both volume and frequency are located on the front panel (see photo 1). The back contains an on/off switch, power socket, female DB25 connector, DIP switch for setting the data rate (75 to 9600 bps), and a miniature speaker jack (see photo 2).

Two cables must be purchased or assembled before you can use the unit. If your headphone set or speaker doesn't have a cable, you will need to obtain one, as well as another cable to connect the DB25 connector on the Type-'N-Talk to the one on your computer. Vodex has 4-foot (1.2 m) cables for most of the popular computers and peripheral cards (retail price $24.95). If they don't have the right set up, the manual includes details of which pins need to be connected. Serial interfaces are set up in a wide variety of ways, so you may need to experiment with your interface to find the right connection pattern. For several examples of how an Apple II computer can be connected, see the textbox entitled "Apple II Interfacing Options."

The serial interface hardware you choose for Type-'N-Talk will be easier to work with if it uses handshaking signals. When the Type-'N-Talk sees a carriage return, the internal microprocessor begins converting the received text to speech. Any text sent while the conversion is being performed will not be properly received. If your hardware doesn't have handshaking, a fix can be done in software by adding a delay loop after each output statement to the Type-'N-Talk in all your programs.

When the unit is successfully connected and powered up, it responds by saying "System ready." If the serial port is represented by device number 2 on the Apple II computer, the BASIC statement

```
PR#2
```

initiates the interface. Information contained in following print statements will be spoken.

At the moment, the easiest way to get Type-'N-Talk is directly from the manufacturer. The price is $375 plus a $4 delivery charge and appropriate sales tax. For more information, contact:

Vodex
500 Stephenson Highway
Troy, MI 48084
1-800-521-1350

Dealerships are being set up around the country so your favorite store may handle them soon.

Summary

I found the Vodex Type-'N-Talk well documented, quite easy to use, and fairly easy to interface to an Apple II. If you have an interest in unlimited vocabulary speech synthesis, this unit provides a powerful, simple, and quick solution. ■

Acknowledgements

Thanks to Russ Mitchell of the Computer Plus store in Sunnyvale, California, for providing the opportunity to try several different interface cards.

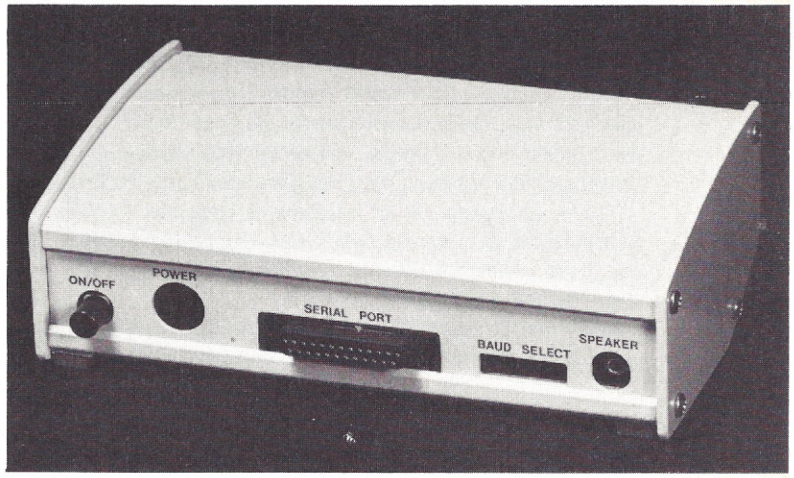


Photo 2: Connections on rear of Type-'N-Talk.

Product Highlights

Name
Type-'N-Talk

Use
Synthesis of speech from text

Manufacturer
Vodex
500 Stephenson Highway
Troy, MI 48084
1-800-521-1350

Price
$375 plus $4 shipping

Dimensions
20 x 13 x 7 cm (8 x 5 x 3 inches)
900 g. (2 lb.)

Features
Unlimited vocabulary; built-in 1 W amplifier; RS-232C interface

Hardware Needed
Interface cable; speaker cable

Hardware Options
Interface cable 4 feet (1.2 m), $24.95

Software Options
Eight control codes

Documentation
32-page manual

Possible Uses
Friendlier programs; output when you can't watch the display; response for children who can't read

Apple II Interfacing Options

The first choice for a serial card for the Vodex Type-'N-Talk might be the Apple High-Speed Serial Interface. This interface, however, does not have any handshaking capabilities, so delay loops must be added after each PRINT statement to ensure that all the information has been spoken before sending additional information. The request to send and clear to send lines (pins 4 and 5 of the DB-25 connector) are connected on the Apple High-Speed Serial Interface, which means that the Apple II always thinks Type-'N-Talk is ready to receive more information. Sending a second line of text while Type-'N-Talk is converting the first line causes confusion, as the unit will then ignore any more information until it is reset. Performing a software delay loop after each line of information is sent to the Type-'N-Talk will solve this problem. Figure 1 shows the necessary connections between the Apple High-Speed Serial Interface and Type-'N-Talk.

The High-Speed Serial Interface is designed to echo all transmitted data to the Apple's 20-character display screen. To avoid this, define an 80-character display using the on-board switches as described in the manual.

California Computer Systems' 7710 A serial interface is quite easy to use with the Type-'N-Talk. Its handshaking ability allows you to write programs without messy delay loops and the interface can transmit information at 9600 bps, the maximum speed at which Type-'N-Talk can accept information. This interface will not echo transmitted text on the Apple's screen. As shown in figure 2, the wires to pins 4 and 5 (request to send and clear to send signals) need to be crossed. This arrangement works equally well for BASIC and Pascal programs.

The easiest way to use the AIO board, made by SSM, is to connect to the terminal connector instead of the modem connector. Figure 3 shows that pins 4, 8, and 20 (request to send, receive line signal, and data terminal ready) are wired together and connected to the data terminal ready signal (pin 20) on the Type-'N-Talk. The AIO interface also has handshaking capability so your programs do not have to be cluttered with unnecessary delay loops. On-board jumpers can be set to transmit at 9600 bps.

The following BASIC statement prevents the AIO board from echoing transmitted data to the Apple's screen:

PRINT " " : POKE 140n,128

where "n" is the slot the interface is plugged into. The POKE statement sets a flag to disable the echo function.

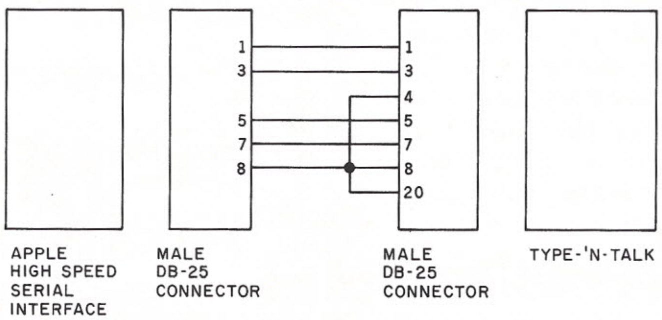


Figure 1: Connections between an Apple High-Speed Serial Interface and Type-'N-Talk.

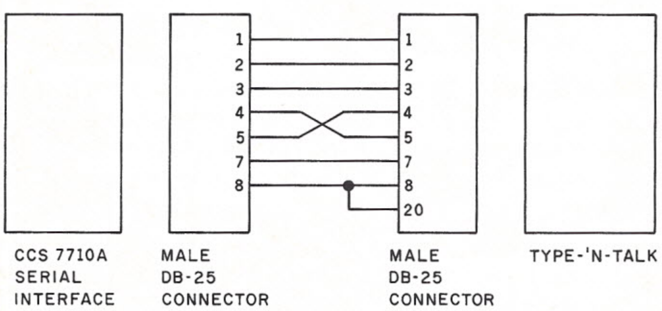


Figure 2: Connections between a California Computer Systems 7710A Serial Interface and Type-'N-Talk.

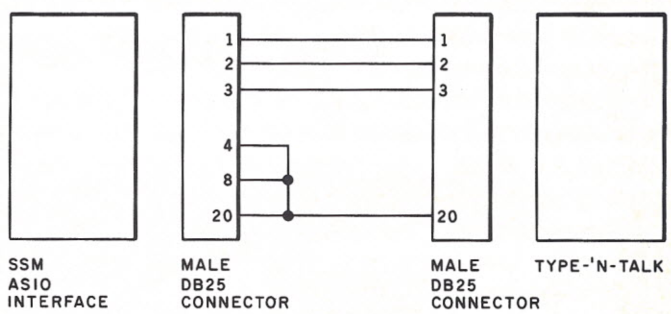


Figure 3: Connections between an SSM ASIO Interface and Type-'N-Talk.

New Products

Design These Steppers Into New Equipment

Two new 15 degree step variable reluctance stepping motors for original equipment manufacturer (OEM) use in computer peripherals and comparable applications have been introduced by Novatronics. Novatronics Size 15, 4 phase, 15 degree VR (variable reluctance) stepping motors are only 1.5 inches long by 1.437 inches in diameter, and are furnished in a choice of two models: 15M34S1 for 12 VDC operation; and 15M34S2 for 24 VDC operation. Both feature a standard size 15 servo motor mounting diameter and groove.

The stepping motors are available with standard plain output shafts, but lead screw output shafts can be supplied on special order. Electrical and mechanical specifications can be tailored to customer requirements.

Prices are quoted according to customer requirements and quantity, and literature on these and other size

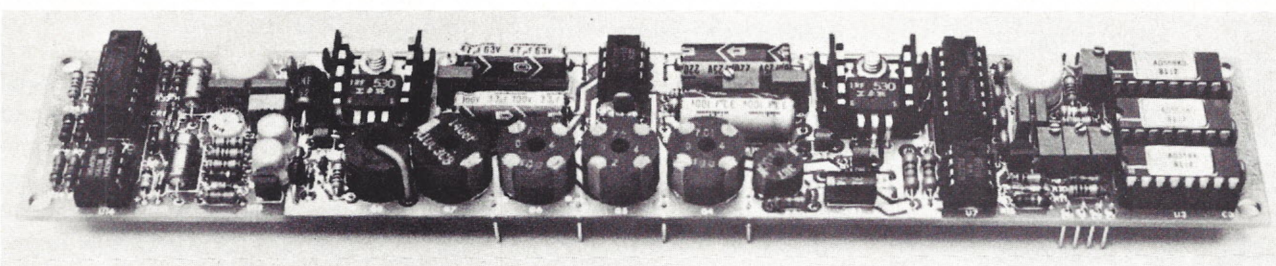
VR motors is available on request. (Inquiries from sales representatives are invited.) For more information contact: Novatronics of Canada Limited, POB 610, Stratford, Ontario, Canada, N5A 6V6, or call (519) 271-3880.
CIRCLE 30

Speech Technology Magazine

Speech Technology (Man/machine voice communications), a quarterly magazine published by Media Dimensions Inc., deals exclusively with the latest advances and applications in voice synthesis and recognition for the engineer, scientist, educator, and manager.

Speech Technology is intended for the engineers and scientists who have wanted to enter this field but have been discouraged by the scarcity of accurate, up-to-date literature in available publications. Likewise, systems and product designers, as well as management, who are interested in learning about and applying the available hardware, will no longer be hampered by a lack of information.

A one-year subscription to *Speech Technology* costs $50. For more information contact: Media Dimensions Inc., 525 East 82nd St., New York, NY 10028, or call (212) 680-6451. CIRCLE 31



A Programmable Power Converter

For those who need exotic onboard power supplies, Interplex, Inc. has introduced the PC77020, a 20 W programmable DC converter.

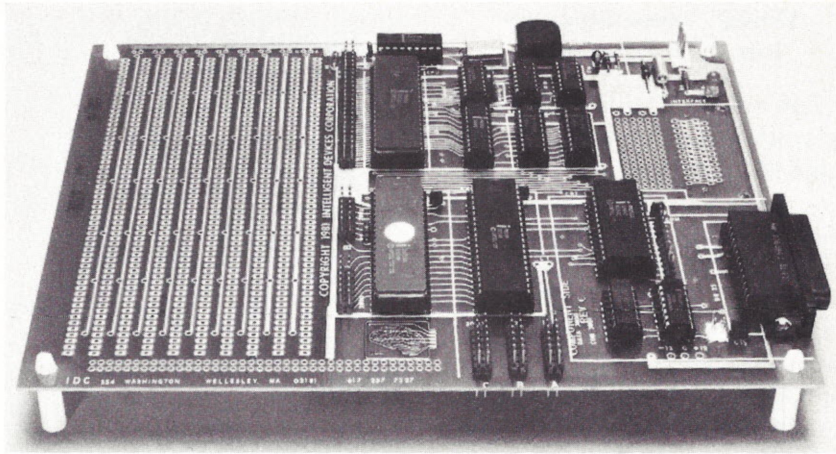
The PC77020 requires a supply of +12 V DC. It can be programmed for an output voltage over a range of 0 to +40.96 V in steps of 10 mV and for a load current over a range of 0 to 2.56 A in steps of 10 mA.

Both voltage and current control loops are continuously active; crossover between constant voltage and constant current control modes is automatic as determined by the more restrictive of either the voltage or current reference at any point in time. A programmed step in voltage is completed in 2 ms.

The PC77020 converter has a low profile with dimensions of 1.5 by 8.875 by 0.4 inches. It is priced at $394 in single quantities; delivery is quoted as six weeks after receipt of order. The company is located at 2680 Bayshore Frontage Rd., Mountain View, CA 94043.
CIRCLE 32

New Products

Need a Self-Contained Subsystem?



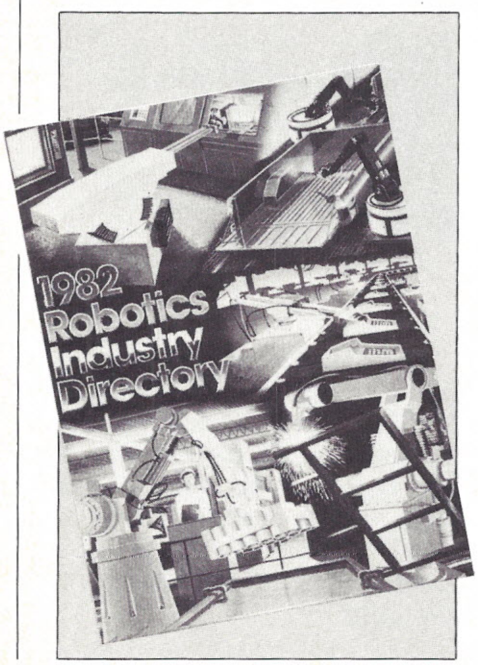
A new single board computer programming and design subsystem, the IDC-8 from Intelligent Devices, is based on Intel's 8088 microprocessor. Software developed for this subsystem is of course compatible with any 8088 or 8086 based computer. One possible use of this system is as a dedicated controller or remote subsystem which is the target of a software development environment based on a larger 8088 or 8086 based computer such as the new IBM Personal Computer.

The IDC-8 comes fully assembled and tested. Its 8088 microprocessor runs at 5 MHz. System components include an 8755 I/O ROM with monitor software, 1K bytes of scratch memory, and 256 bytes of I/O scratch memory. A terminal interface is provided by an 8250 with a standard RS-232C interconnection. Power requirements are 5 V at 1 A, thus you can expect several hours operation off a reasonable battery supply.

Of special note to experimenters with an eye to production, the IDC-8 contains over 18 square inches of wire swap prototype area for special design applications. This area might, for example, be used to support the logic of a robot control interface. The company provides a manufacturing service to convert debugged wire wrap designs in this prototype area into printed circuit versions. They will do this custom manufacturing in quantities of 20 or more.

The IDC-8 has a single unit price of $399, including software and hardware documentation. Delivery is quoted as stock to four weeks. Contact Intelligent Devices Corporation, One Cameron Place, Wellesley, MA 02181.

CIRCLE 33



Industry Directory

With more than one hundred different industrial robot models now available, and prices ranging from $1500 to $250,000, determining which robot is right for a particular application is a formidable task. The 1982 *Robotics Industry Directory* is designed to solve this problem. The *Directory* gives all the information needed to successfully evaluate industrial robots in one easy-to-use, comprehensive volume. Specification matrices are included for quick side-by-side comparisons of competitive models. The *Directory* is indexed, and postpaid Reader Service cards allow you to request further information on any product or service listed.

Included are listings of industrial robots, distributors, consultants and research institutes worldwide, with prices, comprehensive specifications, and the name, address, and phone number of a knowledgeable representative of each company.

The 1982 *Robotics Industry Directory* is priced at $35 in the U.S., $37 in Canada, and $43 foreign. A monthly update is available at additional cost. (With the monthly update the prices are $60, U.S.; $67, Canada; $83, foreign. Foreign prices include airmail postage.) For further information contact: *Robotics Industry Directory*, POB 725, La Canada, CA 91011.

CIRCLE 34

New Products

A New Book of Importance to Robotics Experimenters

Offering thorough coverage of the state of the art in robotics research and design, *Brains, Behavior, and Robotics* by James S. Albus explores the ways in which the brain functions primarily as a computing device for generating and controlling behavior.

This computer-oriented guide provides a detailed assessment of behavior as a product of three hierarchies of computing modules — behavior generating, sensory processing, and memory — which form a world model. The model is examined in relation to known phenomena of learning and conditioning, motor coordination, cognitive development, language skills, planning, dreaming, emotion, and choice. A detailed discussion of artificial intelligence shows the relation of the hierarchical model to vital computer science techniques, including planning, problem-solving, machine vision, natural language understanding, and knowledge representation.

Design considerations in building a robot control system are correlated to the author's model construct, and Albus describes potential uses for robots in industry, in the home, and for space and ocean exploration.

Albus is a project manager of robotics at the National Bureau of Standards in Washington, DC, but the ideas presented in *Brains, Behavior, and Robotics* represent the views of the author and not those of the agency. Formerly affiliated with NASA, Albus was involved in design of subsystems for more than 15 spacecraft, and he managed the NASA artificial intelligence program. *Brains, Behavior, and Robotics* is published by BYTE Books/McGraw-Hill, and is available for $16.95.

CIRCLE 35

Laboratory FORTRAN, Anyone?

It is getting harder to tell the functional difference between a dedicated laboratory instrumentation system and a general-purpose computer with peripherals attached. As evidence we submit this information about a FORTRAN option now available with the Fluke 1720A Instrument Controller.

This self-contained system has a 16-bit microcomputer with a built-in floppy-disk drive and a touch-sensitive display. The new language option is delivered in the form of a floppy diskette. It is priced at $995, which includes a standard FORTRAN software package, a new text editor package, and complete documentation.

The FORTRAN option operates under Fluke's disk-operating system. It is composed of the following programs: FORTRAN Compiler, Linking Loader, Linkage Editor, Library Manager, and a Fluke Library. The Fluke Library is a set of subprograms that can be used to tailor the user's FORTRAN program to the 1720A. Particular attention is given to the IEEE-488 routines to let them resemble the IEEE-488 verbs in the 1720A BASIC. Prominent features of the new language are: sequential and record files, formatted and unformatted I/O, program overlays, double precision and mixed mode arithmetic, relational and logical operators, complete set of IEEE-488 commands, library manager, screen menus, and touch-sense operation.

Computer programs from many computer libraries around the world are compatible with Fluke's FORTRAN IV. In addition, a FORTRAN Matrix Library will soon be available with data storage and retrieval, matrix arithmetic operations, data manipulation, tabulation, and statistics. The product is available from John Fluke Manufacturing Company Inc., POB C9090, Everett, WA 98206.

CIRCLE 36

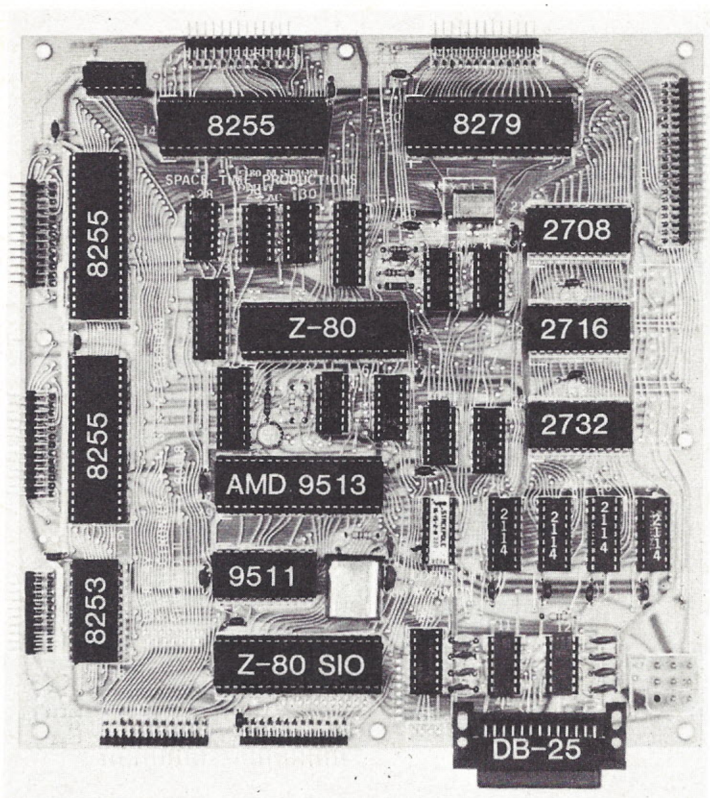
New Products

An Inexpensive Single Board Z-80 Computer

We've received information on a good-looking single board computer product called the "Master Controller Board" which will be an option you should consider in designs requiring a remote or mobile computer. The basic bare board of the MCB product starts at $49.95. Its wiring patterns allow the experimenter to add parts and pieces as required. If fully stuffed with parts, the controller has a 2MHz or 4MHz Z-80 microcomputer, 72 parallel I/O lines through three 8255s, a 66 key keyboard interface through one 8279 decoder, up to 12K bytes of 2732 erasable read only memory (EROM) in three sockets, 2K bytes of scratch memory in 2114 memory parts, one 8253 counter/timer for miscellaneous use, an AMD 9513 for time-of-day clock, two serial ports via a Z-80 SIO chip (one with RS-232C interface and connector), and provision for an AMD9511 or AMD9512 high-speed arithmetic processor. The picture which accompanies the announcement identifies the parts.

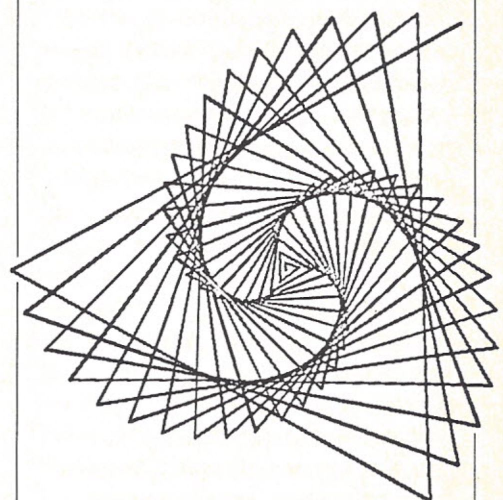
The board, with various parts, is also available in kit form. The "Minimum Kit" at $99.95 includes the processor (Z-80), an 8255, two 2114 parts, a 2708 EROM with test program, and miscellaneous support gates and buffers. (An assembled version is $199.95.) A more elaborate monitor program is available in a 2708 for $39.95, to which a monitor parts kit at $64.95 must be added. Two different power supply modules are also available.

This product is available from R.W. Electronics, 3165 North Clybourn, Chicago, IL 60618.



CIRCLE 37

Explore Interactive Algorithms With Terrapin LOGO



Terrapin, Inc., manufacturers of the robot Turtle, have come out with a version of the LOGO language for the Apple II computer. LOGO is an interpretive language devised to control simulated or real graphic turtles in an educational environment. The language is designed so that young children can easily control the power of computers without having to "know how" to program. Many features common to artificial intelligence research languages are found in LOGO, permitting programs of great power to be written quickly and easily. A brochure outlining the Terrapin LOGO language's features is available.

Suggested retail price of the Terrapin LOGO language for the Apple II computer is $149.95. Every copy of the Terrapin LOGO language includes a disk and documentation. The system requires a 48K Apple II with a 16K memory card or language card, plus one disk drive. A backup diskette is available for $15. Contact Terrapin, Inc., 678 Massachusetts Ave. #205, Cambridge, MA 02139.

CIRCLE 38

New Products

Add 12 New Channels to a Radio Control Link

Vantec has announced their Keykoder systems to expand standard radio control model sets with twelve new on and off commands. This is accomplished by multiplexing one conventional radio control channel; the remaining channels, joysticks and servos of the system operate as before.

The system is comprised of a compact twelve button Keypad mounted on your transmitter and a receptor module that plugs into the radio control receiver like a model servo. The twelve individual outputs from the Receptor each control up to 2.5 A from 4.8 to 28 V. Six of the new functions are momentary-on; the remaining six may be key-on/key-off with memory. Other Receptor configurations are available.

Delivery is quoted as stock to 3 weeks at an introductory price of $289.90 for the complete system. Contact Vantec, 15445 Ventura Blvd., Suite 10-281, Sherman Oaks, CA 91413.

CIRCLE 39

Classified Advertising

HTS KIM/AIM/OSI C1P/VIC-20 PRODUCTS

GENERAL EXPANSION BOARD CONSTRUCTION MANUAL documents two designs, each containing RAM, EPROM, PORTS. Each board occupies a switch-selectable 8K block of memory. Wiring details given for 2K/4K EPROMS, 6522/6532/6821 PORTS, 2114L/6116 STATIC RAMS. Manual — $10. Catalog — Free. Hunter Technical Service; P.O. Box 359; Elm Grove, Wisconsin 53122.

ROBOT R.C. SYSTEM

Transmitter 200 ft. min. 9vdc 10ma. $30.00

Receiver-Open collector to ground output when activated. 12vdc, 20ma. $30.00

Note: Transmitter may be modified for up to 8 channels control. Plans for modification includes easy to use drawings $15.00. Parts for modification will be listed or may be purchased for $12.00. Shipping and Handling charges $3.00. Checks Payable to: RHW Electronics, P.O. Box 2902, Mission Viejo, CA 92690

POSITION OPEN

MECHANICAL AND AEROSPACE ENGINEERING, North Carolina State University, has an opening at the Assistant or Associate Professor level for a person in the specialty area of robotics. The opening is a tenure track full time faculty position. Appointee will be required to teach and develop courses in the area of specialty and participate in and attract sponsored research in support of graduate program effort. Candidate must have a Ph.D. in Mechanical Engineering with training and specialization in robotics. Appointee is expected to have had some teaching experience or be able to demonstrate ability to do so. Must have desire to participate actively in research with demonstrated ability to seek support from outside sources. Some practical industrial experience is highly desirable. Applicants for this position should submit a letter of interest and resume to: Dr. John A. Bailey, Mechanical and Aerospace Engineering, NORTH CAROLINA STATE UNIVERSITY, Raleigh, NC 27650, (919) 737-3024

The University is an Equal Opportunity Employer and operates under Affirmative Action Policy.

ROBOTICS: Unique robot simulation on a computer. Have your own robot for $14.95. Sophisticated programming. Written in Radio Shack Level Two Basic. Send check for cassette to EMPAC, Inc., 439 Edgewood Place, Rutherford, N.J. 07070.

ROBOTICS AGE'S first Assistant Production Manager has first 37 magazines of original issue off press. Will sell for $10 each plus $1.50 for shipping and handling. Includes certified letter stating same. Contact ASBILL, P.O. Box 99 NGC, Tigerville, SC 29688.

Classified Advertising

Robotics Classified Advertisement Order Form

Robotics Age classified advertisements reach thousands of people interested in designing and developing intelligent machines. To place a classified ad simply fill out the order form below and mail to *Robotics Age*, Strand Building, 174 Concord Street, Peterborough, N.H. 03458. The current word rate is $0.50 per word and can be paid by check, MasterCard, or VISA.

All classified advertisements are prepaid and accepted on a first come, first served basis for publication in the next available issue. Publisher reserves the right to reject any advertisement.

Please print:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Word Count: ________ X $0.50 = Total Cost: ________

Name: ______________________________

Company: ______________________________

Street: ______________________________

City/State: ____________________ ZIP: ________

Signature: ______________________________

Payment:
MasterCard ☐ VISA ☐ Check ☐

Card Number: ☐☐☐☐☐☐☐☐☐☐☐☐☐

Interbank Number: ☐☐☐☐

Expiration Date: __/__

The New Robotics Age
Robot Information Service
is coming!

Placing classified advertisements will soon be even simpler with our Robot Information Service. This system will be on-line 24 hours a day with a 300 bps Bell 103A compatible modem. Anyone with a terminal or computer connected to a standard originate modem will be able to call the Robot Information Service and place a classified ad which will be billed to their MasterCard or VISA account. On-line editing allows text to be entered and verified before a final copy is saved for publication. Billing information is provided immediately so you know exactly how much your credit card account will be billed.

Want to start a new subscription or extend a current one? The Robot Information Service will take your orders and bill your MasterCard or VISA account. Changing addresses? The fastest way to tell us your new address is by leaving a message telling us your old address, new address, and subscription number. If you are having any difficulties with your subscription just leave a message and we will get on the problem immediately. No more wondering if that postcard or letter got through to the right person; now everyone with a terminal and 300 bps modem has a direct line to the *Robotics Age* offices.

Interested in the latest *Robotics Age* News? The Robot Information Service can keep you up to date with what's happening in robotics around the world. Fast breaking news, abstracted from information being collected for our various news columns, is provided for everyone. This service will be announced in more detail in a forthcoming issue. Beta testing was begun at the end of January as this issue went to press.

Continued from Page 9

Week, February 1, 1982. "A parade of industry heavyweights is poised to march into the seductive robot market." Cited are two examples, International Business Machines Corporation and Westinghouse Electric Corporation, both of which will soon have products available. Westinghouse recently licensed technology from Italian Olivetti Corporation, and IBM has begun to test market its own robot. In addition, the author states that Bendix, General Motors, Texas Instruments, and United Technologies are planning to enter the robot business.

IBM has taken the wraps off its proprietary design and shipped a dozen robots to selected users for testing. The author says that IBM is moving slowly because, according to a source inside IBM, "we are afraid the response might be heavier than we can handle." IBM could not have a full-scale robot manufacturing operation at its Boca Raton, Florida, facilities before 1983. According to the report, IBM's first product will be "a sophisticated robot with sensors giving the machine's 'fingers' the ability to feel and grip objects."

Business Week also states that Olivetti will initially ship 40 of its Sigma assembly robots to Westinghouse as part of a five-year technical and marketing assistance agreement. Westinghouse will eventually make the Sigma robot itself, and may also produce robots of its own design. The article quotes Anthony A. Masaro, Jr., general manager of the Westinghouse Industrial Automation Division, as saying, "We have some proprietary robot designs. We do not manufacture them now, but that is not to say we won't."

Other industry giants are also positioning themselves for the robot business. Bendix will introduce two robots in March, United Technologies Corporation will market an automatic welder that was developed by a Dutch subsidiary, General Motors Corporation will market a painting robot, and Texas Instruments Inc. is developing a proprietary robot.

Some of the smaller robot manufacturers such as Automatix, Prab Robots, and Unimation must be worried by the entry of the giants, the article says. Westinghouse staffed its new division with 100 engineers, and that group is the offshoot of its Productivity Center (which evaluates new robot applications) with 300 employees. Unimation had revenues of less than $57 million, compared to IBM with 1981 sales of over $25 billion.

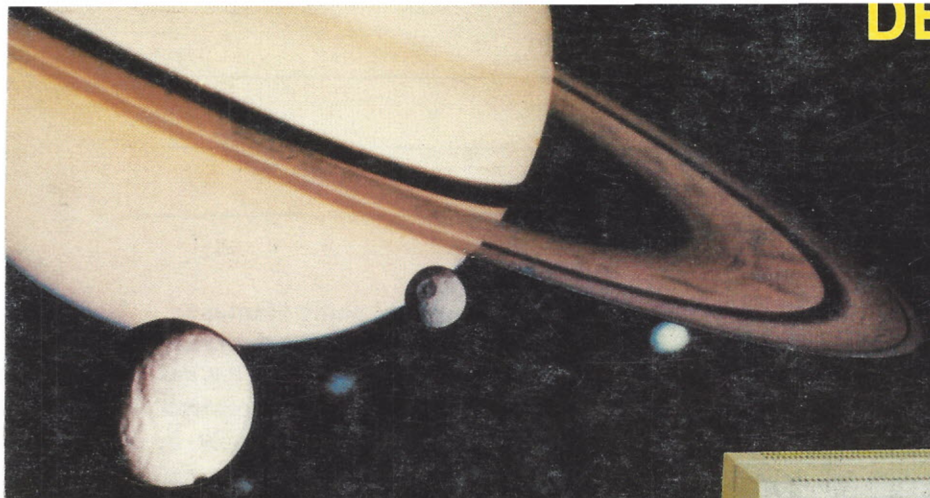
MIT Student Robot Project. A recent airing of the television program *Discover* (which we saw locally on WBZ TV, Boston, on January 26) had a short segment on a robot contest held at the Massachusetts Institute of Technology. The contest was typical of similar student projects that have taken place at MIT. Students were given a standard bundle of materials and a fixed task: produce a machine that would find a hole in the center of a table and place a peg in the hole before a second machine on the table could perform the task. This type of project challenges students to find innovative solutions for complex problems using limited materials.

Providing a Skeptical Outlook on Life. One of the rules for a research scientist or anyone who has to make day-to-day decisions is to keep an open but skeptical mind. All too often we read announcements that describe some fantastic, unbelievable phenomenon which may be "baffling scientists." It is important to view such announcements with more than a pinch of doubt. In robotics, as in any field, such announcements occasionally appear. How are we to distinguish an outlandish claim from one that is a true breakthrough?

A good example of how a fabricated story can appear to be true is the annual April 1st edition of *Scientific American*. Each year a small story appears (usually placed in "Science and the Citizen," "50 and 100 Years Ago," or one of the other small columns) that "sounds" like it could be true. The story is all in fun, and whenever the April issue arrives, my first action is to search out the story.

Douglas R. Hofstader discusses two different approaches to developing a skeptical but open outlook towards new information, in *Scientific American*'s "Metamagical Themes" column (February 1982). In the course of the column he also talks about a publication called *The Skeptical Inquirer*, whose purpose is "to apply common sense to claims of the outlandish, the implausible, and the unlikely." *The Skeptical Inquirer* has been around in one form or another since the fall of 1976. Among the driving forces behind the publication are noted philosophers Ernest Nagel and W. V. Quine, psychologist Ray Hyman, magician James Randi, and mathematician and entertainer Martin Gardner. This periodical should be read by everyone interested in separating truths from shaded innuendo. For more information, write *The Skeptical Inquirer*, Box 229, Central Park Station, Buffalo, NY 12425.

DESKTOP 68000 + TELESOFT[1] ADA[4] + PASCAL = POWER



COMPATIBILITY WITH DEC Q-BUS[3] AND STANDARD DEC[3] PERIPHERALS

THE IN/7000D BRINGS OUT OF THIS WORLD A/I PERFORMANCE WITHIN REACH

68000 BASED IN/7000D
Dramatically increased processing speed and flexibility:
- 8 MHz 68000 CPU
- 32-Bit internal arithmetic registers
- 24-Bit address register
- Powerful assembly language instructions support modular programming
- Designed for expansion to 32 bit word size
- 8 Levels of interrupt priority
- Vectored interrupts and DMA fully supported
- Outbenchmarks the IBM 370/145[2]*
- High speed string processing

Vastly increased memory:
- 68000 directly addresses 16 MB of memory
- Sorts can be done in core rather than with disk I/O

PASCAL AND TELESOFT[1] ADA[4]
- Fast program development
- Self documenting
- Supports structured programming
- Easy to update
- Easy to maintain
- Transportable from computer to computer
- Powerful logical constructs greatly simplify programming

For example: Modular procedures and functions
Strong data types
Case structures
- Local and global variables
- Recursive problem solving
- Block insert — a block of statements may be inserted anywhere one statement can exist
- Built in Boolean functions
- Library capability
- Program segmentability
- Procedure linking
- TeleSoft[1] Pascal and TeleSoft[1] Ada[4] translate to 68000 native code
- Built-in powerful string-handling features

TELESOFT[1] ADA[4]
- Designed to fulfill all DoD specifications
- INTRINSIC functions include:
  - * Multi-tasking and multi-programming
  - * Independent compilation of program units (called Packages)
- Fully implemented syntax checker which now parses the entire Ada[4] language

Saves development time.
- High speed sensory data processing
- High speed string processing power
- Fast coordinate transformations
- Easy implementation of in-memory AI algorithms, predicate calculus and trajectory computations
- Design and test algorithms quicker and easier
- Both Pascal and 68000 support features that make debugging far more efficient
- Plenty of memory, no need to use extra time for "programming tricks," previously needed with limited memory
- Mixed mode listing (Pascal source statements followed by 68000 statements)

UP TO 4 MByte OF RAM

*"Kilobaud Microcomputing" October, 1980
[1]A trademark of Renaissance Telesoftware Inc.
[2]A trademark of International Business Machines
[3]A trademark of Digital Equipment Corporation
[4]A trademark of Department of Defense

Call us now for more information, or fill in the coupon 301/984-8000

I AM INTERESTED! Send more information now.

Name ______________ Title ______________
Company ______________
Street ______________
City ______________ State ______ Zip ______
Phone ______________ Call me IMMEDIATELY ☐
My need is: ☐ 1-3 mo. ☐ 3-6 mo. ☐ Future project ☐ File
☐ I am interested in a dealership.

INTELLIMAC/Intelligent Machines
6001 Montrose Road • Sixth Floor • Rockville, MD 20852
Phone: 301/984-8000
TWX: 710-828-9800 Cable: INTELLIMAC

INTELLIMAC
Intelligent Machines

6001 Montrose Road
Sixth Floor
Rockville, MD 20852
Phone: 301/984-8000
TWX: 710-828-9800
Cable: INTELLIMAC

Dealerships are available

CIRCLE 6

SATURN PHOTOGRAPH COURTESY OF NAS